Handbuch IT in Bibliotheken

15.11.22

Dieses Handbuch gibt einen knappen, umfassenden Überblick über die wesentlichen IT-bezogenen Themen in Bibliotheken. Die aktuelle Version beschränkt sich auf Kapitel zum Management von IT-Systemen, Anforderungen an die IT-Entwicklung und Bibliotheksmanagementsysteme.

Inhaltsverzeichnis

# Einleitung

## Motivation

Because the library has become software, it is no longer viable for our services to exist separately from our software. [...] Most importantly, all library staff must understand that our software is our library, and is everyone’s responsibility.

— (Hanson, Cody 2015)

Mit der wachsenden Bedeutung der Informationstechnologie (IT) im Allgemeinen und für Bibliotheken im Besonderen bleibt kaum ein Aspekt bibliothekarischer Aufgaben, der nicht durch IT unterstützt wird. Deutlich wird dies z.B. durch das stetig zunehmende Angebot elektronischer Informationsmittel, die Digitalisierung historischer Bestände, interoperable Metadaten oder auch Langzeitarchivierung. Die alltägliche Handhabung von IT (Smartphones, Automatisierung, Vernetzung…) wird häufig einfacher, die zugrundeliegenden Systeme werden jedoch immer komplexer und erfordern entsprechend mehr Wissen zu ihrem Aufbau und Betrieb. Während sich einige Teile der IT in Bibliotheken nicht wesentlich von IT in anderen Bereichen unterscheiden, gibt es doch zahlreiche Aspekte von Bibliotheks-IT, die nicht oder nicht speziell genug an anderer Stelle behandelt werden, beispielsweise die zentralen [Bibliotheksmanagementsysteme](#bibliotheksmanagementsysteme).

## Inhalt

Dieses Handbuch soll einen knappen, umfassenden Überblick über die wichtigsten IT-bezogenen Themen in Bibliotheken geben. Die vorliegende Version behandelt:

* [Management von IT-Systemen](#management)
* [Anforderungen an die IT-Entwicklung](#anforderungen)
* [Bibliotheksmanagementsysteme](#bibliotheksmanagementsysteme)

Weitere Themen werden bis zur Vervollständigung des Handbuchs hinzukommen.

## Über dieses Handbuch

Die erste Version dieses Handbuchs wurde [im Rahmen eines Book Sprints](https://www.th-wildau.de/book-sprint/) im April 2022 an der Bibliothek der Technischen Hochschule Wildau erstellt. Dazu trafen sich zwölf IT-affine Expert\*innen aus dem Bibliotheksbereich, um innerhalb von drei Tagen eine umfassende Übersicht speziell zum Thema Bibliotheksmanagementsysteme (BMS) zu verfassen. Weitere Autor\*innen wurden online eingebunden. Die Veranstaltung wurde mit Mitteln des [Publikationsfonds für Open-Access-Monografien des Landes Brandenburg](https://open-access-brandenburg.de/fonds/) gefördert.

Zur Klärung der [Zielgruppe](#zielgruppe) wurden einige sogenannte *Personas* definiert. Eine Brainstorming-Session ergab zunächst relevante Themen für den Teilbereich [Bibliotheksmanagementsysteme](#bibliotheksmanagementsysteme) und eine grobe Gliederung in Unterkapitel. Diese wurden anschließend in mehreren Iterationen gemeinsam ausformuliert.

Im Laufe des Schreibprozesses wurden allgemeine Aspekte des [Management von IT-Systemen](#management) und [Anforderungen an IT-Entwicklung](#anforderungen) in eigene Kapitel ausgegliedert. Weitere Themen wie IT-Entwicklung, Forschungsunterstützung und Datenformate sind bisher nicht oder nur fragmentarisch behandelt. Um diese Kapitel zu vervollständigen und weitere Kapitel anzulegen, sollen weitere Book Sprints veranstaltet werden.

Das Handbuch ist ein „lebendiges Buch“, das stetig ergänzt und aktualisiert werden kann. Bei der Bearbeitung sollten folgende Grundsätze beachtet werden:

* Wir verzichten auf individuelle Autorenschaft an einzelnen Textteilen. Alle können an allen Teilen mitarbeiten.
* Das Handbuch ist keine wissenschaftliche Arbeit, sondern soll einen Überblick geben. Für Details kann auf weiterführende Quellen verwiesen werden.

Kontaktinformationen und Neuigkeiten zum Projekt zur Erstellung des Handbuchs finden sich auf der Seite <https://www.th-wildau.de/book-sprint/>.

|  |
| --- |
| Tipp |
| Weitere [Hinweise zur Mitarbeit](#mitarbeit) und Details zur technischen Umsetzung des Handbuchs finden sich im Anhang. |

## Beteiligte Autor\*innen

* Nicolas Bach, Student an der HdM Stuttgart  
  [nicolas.bach@posteo.de](mailto:nicolas.bach@posteo.de)
* Janna Brechmacher, Stabstelle IT in der Benutzungsabteilung der Staatsbibliothek zu Berlin  
  <https://orcid.org/0000-0001-7233-7153> [janna.brechmacher@sbb.spk-berlin.de](mailto:janna.brechmacher@sbb.spk-berlin.de)
* Sascha A. Carlin, Agile Coach für Führungskräfte in der Softwareentwicklung  
  [rqst@nvsbl.cm](mailto:rqst@nvsbl.cm)
* Anne Christensen, Bibliothekarin und Projektmanagerin bei effective WEBWORK sowie Lehrbeauftragte an verschiedenen Hochschulen  
  <https://orcid.org/0000-0001-7753-1078> [christensen@effective-webwork.de](mailto:christensen@effective-webwork.de)
* Jana Eger, Stadtbibliothek Chemnitz  
  [jana.eger@stadtbibliothek-chemnitz.de](mailto:jana.eger@stadtbibliothek-chemnitz.de)
* Gerrit Gragert, Leitung IT-Services für die Digitale Bibliothek in der Staatsbibliothek Berlin  
  <https://orcid.org/0000-0002-0542-1555> [gerrit.gragert@sbb.spk-berlin.de](mailto:gerrit.gragert@sbb.spk-berlin.de)
* Lambert Heller, Leitung Open Science Lab an der TIB - Leibniz‐Informationszentrum Technik und Naturwissenschaften  
  <https://orcid.org/0000-0003-0232-7085> [lambert.heller@tib.eu](mailto:lambert.heller@tib.eu)
* Sina Hurnik, Studio für Kommunikationsdesign  
  <https://sinahurnik.com/>
* Clemens Kynast, Discoverysysteme & Bibliotheksautomatisierung an der ThULB Jena  
  [clemens.kynast@uni-jena.de](mailto:clemens.kynast@uni-jena.de)
* Lukas Lerche, UB Dortmund  
  <https://orcid.org/0000-0002-4027-6840> [lukas.lerche@tu-dortmund.de](mailto:lukas.lerche@tu-dortmund.de)
* Luis Moßburger  
  <https://orcid.org/0000-0002-5326-219X> [lmossburger@t-online.de](mailto:lmossburger@t-online.de)
* Michael Schaarwächter, Bibliotheks-IT an der UB Dortmund  
  <https://orcid.org/0000-0002-0180-5930> [michael.schaarwaechter@tu-dortmund.de](mailto:michael.schaarwaechter@tu-dortmund.de)
* Frank Seeliger, Bibliotheksleiter TH Wildau  
  <https://orcid.org/0000-0003-0602-8082> [fseeliger@th-wildau.de](mailto:fseeliger@th-wildau.de)
* Jakob Voß, Forschung und Entwicklung an der VZG Göttingen  
  <https://orcid.org/0000-0002-7613-4123> [jakob.voss@gbv.de](mailto:jakob.voss@gbv.de)
* Michael Voss, Consultant; IT-Expert-Voss  
  <https://orcid.org/0000-0002-7402-1598> [info@it-expert-voss.de](mailto:info@it-expert-voss.de)
* David Zellhöfer, Professor für Digitale Innovation in der öffentlichen Verwaltung an der HWR Berlin  
  <https://orcid.org/0000-0002-0403-457X> [david.zellhoefer@hwr-berlin.de](mailto:david.zellhoefer@hwr-berlin.de)

## Rechte an den Inhalten des Buchs

Soweit nicht anders in Quellenangaben ausgewiesen, stehen alle Inhalte dieses Buches unter der Lizenz [Creative Commons Namensnennung 3.0 Deutschland](https://creativecommons.org/licenses/by/3.0/de/) (CC-BY 3.0 DE).

Das heisst: Sie dürfen das Material in jedwedem Format oder Medium vervielfältigen und weiterverbreiten (Teilen) und das Material remixen, verändern und darauf aufbauen (Bearbeiten) und zwar für beliebige Zwecke, inklusive kommerzielle Zwecke, unter der Bedingung, dass Sie angemessene Urheber\*innen- und Rechteangaben machen, einen Link zur Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden (Namensnennung).

# Management von IT-Systemen

|  |
| --- |
| Hinweis |
| IT-Systeme sind selten statisch sondern folgen einem [Lebenszyklus](#lebenszyklen-von-it-systemen) von der Planung bis zu ihrer Ablösung. Während des Betriebs der Systeme müssen mögliche [Risiken](#betriebssicherheit-und-risikomanagement) beachtet und [rechtliche Rahmenbedingungen](#rechtliche-rahmenbedingungen) eingehalten werden. In Bibliotheken sind daher entsprechende [IT-Kompetenzen](#kompetenzen) und ein [organisatorischer Rahmen](#organisation) notwendig. Um diesen Anforderungen begegnen zu können, gibt es Möglichkeiten zur [Aus- und Weiterbildung](#aus--und-weiterbildung). |

Dieses Kapitel konzentriert sich auf den Einsatz von IT-Systemen in Bibliotheken. Ein Kapitel zur IT-Basis-Infrastruktur (Netzwerke, Netzdienste, Server, Storage, Backup u.ä.) ist in Planung.

## IT-Systeme

### Lebenszyklen von IT-Systemen

Alle Software-Systeme folgen einem Lebenszyklus, der mit ihrer Einführung beginnt und irgendwann mit ihrer Abschaltung endet ([Abbildung 1.1](#fig-it-zyklus)). Die wesentlichen Phasen im klassischen Lebenszyklus eines IT-Systems werden im Folgenden näher betrachtet.

Die konkrete Abfolge vor allem der ersten Phasen kann je nach der angewendeten Projektmanagement-Methode (agil vs. klassisch) variieren. Eine Diskussion von agilen und klassischen Methoden liegt außerhalb des Fokus dieses Handbuchs.

|  |
| --- |
| Abbildung 1.1: SDLC-Skizze (Platzhalter) |

#### Planung und Analyse

Grundlage für die Umsetzung eines Softwareprojekts, egal ob es sich um individuell erstellte Software oder die Anpassung eines existierenden IT-Systems handelt, ist ein gemeinsames Verständnis für das Ziel und die Anforderungen des Projektes. Dieses gemeinsame Verständnis, insbesondere der Anforderungen, sollte bei allen Projektmitgliedern und den weiteren Stakeholdern vorhanden sein.

Zur Planungs- und Analysephase gehört neben einer grundsätzlichen Machbarkeitsanalyse des Projekts die Zusammenstellung eines geeigneten Teams, die Bestimmung der Stakeholder sowie die Klärung finanzieller und rechtlicher Rahmenbedingungen.

|  |
| --- |
| Tipp |
| Zuweilen kommt es vor, dass die Entscheidung für ein IT-System bereits getroffen ist, bevor geklärt wurde, welches Problem damit gelöst werden soll. Auch in diesem Fall ist es sinnvoll, die Einführung mit einer offenen Planung und Anforderungsanalyse zu beginnen, und danach zu prüfen, welche Anforderungen das System tatsächlich abdecken kann. |

#### Design/Prototyping

Während der Design- bzw. Prototyping-Phase entwickeln Designer\*innen und Entwickler\*innen erste Prototypen. Ziel ist es dabei, Feedback der verschiedenen Stakeholder zu erhalten, um gemeinsam ein besseres Verständnis der Anforderungen zu erhalten bzw. diese zu präzisieren. Das Kapitel [Anforderungen an Bibliotheks-IT](#anforderungen) geht gesondert auf die Bedeutung dieser Einbeziehung und damit verbundener Methoden ein.

#### Implementierung

Aufbauend auf einem gemeinsamen Verständnis der Anforderungen überführen Entwickler\*innen Prototypen in lauffähigen Code. Wird im Rahmen des Projekts ein bestehendes System implementiert, werden die Prototypen zunächst in ein Testsystem und in der Folge in das produktive System überführt.

In klassischen Projekten sieht man in dieser Phase zuerst ein Produkt mit den gewünschten Features, während nutzer\*innenorientierte Vorgehensmodelle (siehe Kapitel [Anforderungen](#anforderungen)) hier auf einen iterativen Prozess setzen, welcher Produktiterationen häufiger bereitstellt und evaluiert.

#### Test und Integration

Als letzte Lebensphase vor der Produktivschaltung werden Abnahmetests und die Integration des entwickelten bzw. erworbenen Systems in die Zielumgebung durchgeführt. Im Falle der Inanspruchnahme eines Dienstleisters wird hier auch dessen Leistung final abgenommen, wenn das System erfolgreich produktiv in Betrieb genommen werden kann.

Während der Tests wird korrekte Umsetzen der Anforderungen sowie die Umsetzung der Anforderungen geprüft.

#### Wartung

Die Wartungsphase folgt auf die Produktivsetzung des IT-Systems. In dieser Lebensphase wird das System nicht mehr grundlegend weiterentwickelt, es werden jedoch Fehler (Bugs) entfernt und Anpassungen der Funktionsweise im Sinne der Parametrisierung oder die Optimierung der Programmabläufe vorgenommen.

Typischerweise finden sich IT-Systeme, die grundlegende Geschäftsprozesse abbilden oder die nach individuellen Anforderungen erstellt wurden, viele Jahre in dieser Phase. [Abbildung 1.2](#fig-verweildauer) illustriert die Lebensspanne einiger ausgewählter Nachweissysteme der Staatsbibliothek zu Berlin, die zum Zeitpunkt der Erstellung dieses Textes erst teilweise abgelöst wurden.

|  |
| --- |
| Abbildung 1.2: Lebenszeit (in Jahren) von Bibliothekssystemen in der Wartungsphase am Beispiel der Staatsbibliothek zu Berlin (Stand 2022) |

#### Ablösung

Die Ablösung eines Systems kann eine Vielzahl an Gründen haben. So entwickeln sich die technischen Möglichkeiten und die Anforderungen der Nutzer\*innen kontinuierlich weiter. Eine Ablösung kann aber auch durch technische Obsoleszenz erzwungen werden, wenn zugrundeliegende Software-Komponenten wie das Betriebssystem oder ein Datenbankmanagementsystem nicht mehr sicher betrieben werden können.

Die konkrete Ablösungsplanung sollte mit genügend zeitlichem Vorlauf begonnen werden. Dies gewährleistet die Arbeitsfähigkeit in der Ablösungsphase. So können in der Vorphase beispielsweise notwendige Daten migriert werden, die vom Altsystem vorgehalten werden.

Mit dem frühzeitigen Beginn der Ablösungsplanung noch in der Wartungsphase können zudem vermeidbare Risiken minimiert werden. Dies ist insbesondere deshalb wichtig, da man als Betreiber eines IT-Systems nicht alle Faktoren kontrolliert, welche eine kurzfristig notwendig werdende Ablösung des Systems verursachen können. Darunter fallen zum Beispiel:

* die Abschaltung wegen technischer Obsoleszenz (s.o.),
* der Ausfall des Systems durch Hardware-Ausfälle,
* die Ankündigung von Wartungsarbeiten und Sicherheits-Patches durch den Hersteller oder die Insolvenz des Herstellers (insb. bei proprietärer Software) oder
* die De-Facto-Unwartbarkeit durch den Wegfall geeigneten Personals mit Spezialkenntnissen (z.B. veralteter Programmiersprachen), siehe dazu auch den Abschnitt [Ressourcenplanung](#ressourcenplanung)

Letztlich führen all diese Punkte zur Abschaltung eines IT-Systems aufgrund von IT-Sicherheitsproblemen, da diese Einbrüche in die Systeme (Hacks) begünstigen. Hinzu kommt das Risiko von Datenverlusten, entweder durch physischen Verlust im Falle eines Hardware-Defekts oder durch den logischen Verlust, da z.B. proprietäre Datenformate nicht mehr gelesen werden können.

Der Weiterbetrieb eines IT-Systems ohne Ablösungsplanung birgt hohe Risiken in sich und kann eine Organisation folglich in ernsthafte Schwierigkeiten bringen, insbesondere wenn geschäftskritische Prozesse betroffen sind.

### Betriebssicherheit und Risikomanagement

Neben den Problemen der Ablösungplanung gibt es weitere Risiken des Betriebs von IT-Systemen, von denen einige im nachfolgenden Abschnitt vorgestellt werden.

#### Vendor-Lock-In

Ein nicht zu unterschätzendes Risiko, welches sich aus der Einführung eines proprietären IT-Systems ergibt, ist der sogenannte Vendor Lock-In. Dieser beschreibt die Abhängigkeit von Produkten oder Dienstleistungen eines Anbieters durch die der gleichzeitige Einsatz von anderen Produkten oder der Wechsel zu anderen IT-Systemen erschwert wird. Durch den Einsatz von Systemen mit etablierten Standards, offenen Datenformaten und Schnittstellen sowie geeigneter Ablösungsstrategien kann das Risiko eines Vendor Lock-Ins verringert werden.

Der Begriff des Vendor Lock-ins kann noch auf den Bereich der Fehlerbehebung und die Wartung von Software ausgedehnt werden. Im Fall von proprietärer Software, welche ohne Zugriff auf den Quellcode betrieben wird, ist die Fehlerbehebung ausschließlich Sache des Herstellers. Fällt dieser, wie oben beschrieben, aus, kann ein Betrieb aus IT-Sicherheitsperspektive nicht mehr verantwortet werden. Hinzu kommt, dass das sogenannte Reverse Engineering bzw. das Dekompilieren dieser Software in der Regel verboten ist. Mit einem [Grundsatzurteil des EuGH aus dem Jahr 2021](https://curia.europa.eu/juris/document/document.jsf?text=&docid=247056&pageIndex=0&doclang=DE&mode=lst&dir=&occ=first&part=1&cid=9912038) wird dieses Verbot jedoch aufgeweicht. So ist es nun rechtmäßigen Erwerbern erlaubt, Fehler in einem Computerprogramm zu beheben und dafür auch proprietäre Software zu dekompilieren.

In der Praxis sollte dieses Notfallszenario aber nicht in die Planung einbezogen werden, da die Fehlerbehebung innerhalb fremder Software unter dem Rückgriff auf Dekompilierung besondere Kenntnisse seitens des zuständigen IT-Personals voraussetzt.

#### Software-Abhängigkeiten

Sowohl der Betrieb von proprietärer als auch von Open-Source-Software ist vom Funktionieren einer Vielzahl weiterer Software-Komponenten abhängig. Diese Abhängigkeit lässt sich mit einem vereinfachten Schichtmodells des Betriebs eines IT-Systems illustrieren:

|  |
| --- |
| Abbildung 1.3: Schichtmodell-Bild (Platzhalter) |

Aus [Abbildung 1.3](#fig-schichtmodell) wird deutlich, dass moderne Software-Systeme zum Beispiel auf einem Betriebssystem oder weiteren Subsystemen wie einem Datenbankmanagementsystem basieren. Um das gesamte IT-System betreiben zu können, müssen die Einzelkomponenten zusammen spielen. Fällt eines der Systeme, beispielsweise das Betriebssystem, aufgrund von Obsoleszenz aus, so ist es unter Umständen möglich, die darüber liegenden Schichten auf ein neues Betriebssystem zu migrieren, jedoch ist dies nicht garantiert.

Das Risiko erhöht sich, wenn im Rahmen eines Wartungsvertrags durch den Hersteller festgelegt wurde, dass zum Beispiel nur bestimmte Kombinationen aus Betriebssystem und weiterer Komponenten zugelassen sind. In diesem Fall kann ein IT-System aus der Wartung fallen, obwohl es vorerst betreibbar bleibt. Mit dem Ausfall der Wartung entfallen auch Software-Updates etc. Damit ist der mittel- bis langfristige Weiterbetrieb des Systems ohne Gefährdung der Betriebssicherheit aller IT-Systeme der Organisation nicht möglich.

Das Kapitel [IT-Entwicklung] widmet sich dem Management von Software-Abhängigkeiten im Detail.

### Rechtliche Rahmenbedingungen

Die meisten Bibliotheken befinden sich in öffentlicher Hand und sind deshalb bestimmten Gesetzen und Verordnungen unterworfen. Von besonderer Bedeutung sind dabei Anforderungen an die Software-Ergonomie und die Barrierefreiheit (Accessibility) von IT-Systemen.

#### Software-Ergonomie

Die gesetzliche Unfallversicherung fordert z.B. die Berücksichtigung ergonomischer Grundsätze bei der Entwicklung von Software. Moderne grafische Anwendungen müssen ebenso wie Internetseiten diese Anforderungen erfüllen:

Die Software muss gebrauchstauglich sein, das heißt, sie sollte gewährleisten, dass Benutzer festgelegte Ziele in einem bestimmten Nutzungskontext effektiv, effizient und zufriedenstellend erreichen können. Dies setzt voraus, dass die Grundsätze der Dialoggestaltung nach *DIN EN ISO 9241-110*, wie *Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Steuerbarkeit, Fehlertoleranz, Erwartungskonformität, Individualisierbarkeit, Lernförderlichkeit* beachtet und realisiert werden.

– (Gesetzliche und Unfallversicherung e.V. (DGUV) 2019)

Die Erreichung dieser Ziele wird im Kapitel [Anforderungen an IT-Entwicklung](#anforderungen) thematisiert.

#### Barrierefreiheit

Neben dem Befolgen der Anforderungen an ergonomisch bedienbare Software, liegt es auf der Hand, dass IT-Systeme für eine Vielzahl von Anwender\*innen nutzbar sein sollte. Diese grundlegende Anforderung bezeichnet man als Barrierefreiheit bzw. Accessibility.

Während Barrierefreiheit häufig mit einem sehr engen Behinderungsbegriff assoziiert wird, wie z.B. die Rampe für Rollstuhlfahrer\*innen, ist dieser Begriff mittlerweile aufgrund der gesetzlichen Grundlagen in Deutschland wesentlich weiter zu fassen (siehe [§3 Behindertengleichstellungsgesetz](https://www.gesetze-im-internet.de/bgg/__3.html)). So leiten sich z.B. auch Anforderungen an Angebote in Leichter Sprache o.ä. aus diesem weiten Behinderungsbegriff ab.

Aufbauend auf der einschlägigen Gesetzgebung regelt die [BITV 2.0](https://www.gesetze-im-internet.de/bitv_2_0/index.html) (Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz) die konkrete Gestaltung barrierefreier IT-Systeme und Webangebot. Hierbei greift sie auf die aktuell gültigen Web Content Accessibilty Guidelines (WCAG) zurück, welche die Anforderungen der Barrierefreiheit anschaulich mit vielfältigen Beispielen illustriert.

Die gesetzliche Anforderung Barrierefreiheit umsetzen zu müssen trifft dabei nicht nur auf Organisationen der öffentlichen Verwaltung, wie es viele Bibliotheken sind, zu. Vielmehr müssen sich alle Stellen, die europäisches Vergaberecht anwenden müssen (siehe EU Richtlinie 2016/2102), z.B. im Rahmen von Drittmitteln oder Zuwendungen, nach diesen Vorgaben richten.

Wichtig ist hierbei zu beachten, dass die BITV 2.0 nicht zwischen internen und externen Nutzer\*innen unterscheidet. Das heißt, dass sowohl rein bibliotheksintern genutzte Systeme als auch nach außen gerichtete Anwendungen, wie z.B. Discovery-Systeme, barrierefrei zu gestalten sind.

Neben der Ermöglichung der digitalen Teilhabe für den Großteil der Bevölkerung sollen abschließend noch drei weitere positive Aspekte der Beachtung von Barrierefreiheitsanforderungen genannt werden:

1. Viele Umsetzungen der Grundsätze der Barrierefreiheit, erhöhen auch die Usability für Menschen ohne Behinderung,
2. ebenso wird zumeist die Nachnutzbarkeit auf Mobilgeräten erhöht und
3. die Gestaltung barrierefreier Anwendungen schlägt sich positiv im Suchmaschinenranking nieder, da [Suchmaschinen diesen Aspekt zur Bewertung heranziehen](https://developers.google.com/search/blog/2020/05/evaluating-page-experience).

## Management der Bibliotheks-IT

Die Auswahl und Implementierung sowie der Betrieb von digitalen Diensten ist ein stetig wachsender Aufgabenbereich für Bibliotheken. Durch den Verlust ihres früheren Monopols auf die Versorgung mit Informationen ist Ende der 1990er Jahre in den Bibliotheken ein starker Innovationsdruck entstanden. In der Folge wurden neue Dienstleistungen wie fachliche Portale, Dokumentenserver etc. im Rahmen von Projekten realisiert. Die notwendigen Kenntnisse haben vielerorts eigens dafür eingestellte Mitarbeiter\*innen eingebracht. Eine systematische Ausweitung von Kenntnissen zu IT-Systemen, Metadatenmanagement, Web-Standards, [Usability und User Experience] bleibt jedoch für die meisten Bibliotheken eine große Herausforderung.

### Kompetenzen

Der Einsatz von IT-Systemen erfordert dezidierte Kenntnisse und Fähigkeiten. Veränderungen der Systemlandschaft, z.B. durch einen Systemumstieg oder die Einführung neuer Systeme, erfordern daher eine regelmäßige Analyse der vorhandenen und benötigten IT-bezogenen Kompetenzen.

Bei der Analyse sind folgende Bereiche zu unterscheiden:

* Basisinfrastruktur für Hard- und Software: Infrastruktur Netze/Hardware, Netzdienste, Server, Basis-Software, Storage, Backup
* Systemadministration: Installation und Betrieb
* Inbetriebnahme und Individualisierung: Metadatenmanagement, Konfiguration und Parametrisierung, Software-Entwicklung
* Projektmanagement

Für die Systemadministration werden Kenntnisse zu grundsätzlichen Architekturen webbasierter Systeme benötigt. In der Regel handelt es sich dabei um die Installation und Administration von Datenbanken, PHP- und Java-Systemen auf Linux-Servern, einschließlich der jeweiligen Wartung durch Minor und Major Updates, Patches etc.

Für die Inbetriebnahme und Individualisierung von Systemen sind zunächst Kenntnisse bei der Umwandlung von Daten in unterschiedliche Formate notwendig, wenn eine Datenmigrationen aus Altsystemen durchgeführt wird. Die Systeme müssen dann konfiguriert und parametrisiert werden, das heißt auf die konkreten Nutzungsszenarien angepasst. Hier ist zwischen einer so genannten Out-of-the-Box-Verwendung, bei der das System im Rahmen seiner Möglichkeiten genutzt wird, und einer weitergehenden Individualisierung durch eigene Software-Entwicklung zu unterscheiden. In beiden Fällen ist ein Zusammenspiel von bibliotheksfachlichen und Software-technischen Kompetenzen erforderlich, um das Verständnis von bibliothekarischen Geschäftsgängen und den Prozessen und Funktionalitäten des Systems zusammen zu bringen. Daher werden in der Regel Implementierungsteams aus Anwender\*innen und Software-Betreuer\*innen bzw. Entwickler\*innen gebildet

Die Arbeit dieser Implementierungsteams sollte idealerweise nach Grundlagen des Projektmanagements und des [Lebenszyklus von IT-Systemen](#Lebenszyklen-von-IT-Systemen) erfolgen, also unter Berücksichtigung klarer Strukturen für die Planung, die Kommunikation und die Kontrolle. Siehe dazu auch das Kapitel [IT-Entwicklung].

|  |
| --- |
| Tipp |
| Niemand kommt als IT-Expert\*in auf die Welt und es ist praktisch unmöglich bei allen Entwicklungen auf dem Laufenden zu bleiben. Versuchen Sie ihre Kompetenzen realistisch einzuschätzen und scheuen Sie sich nicht Kolleg\*innen um Rat zu fragen! |

### Organisation

Größere Bibliotheken haben in der Regel eigene IT-Abteilungen, die folgende Aufgaben übernehmen:

* Anwendungsbetreuung bei Hard- und Software sowie Peripheriegeräten,
* Betreuung von eigenen Servern,
* Konfiguration und Parametrisierung von Systemen,
* ggf. Software-Entwicklung.

Alle Punkte außer der Konfiguration und Parametrisierung von Systemen werden in der Regel von informatisch ausgebildetem Personal vorgenommen. Die Konfiguration und Parametrisierung von Systemen ist klassischerweise eine Aufgabe so genannter Systembibliothekar\*innen, die teilweise in Ergänzung zu ihrer bibliothekarischen Ausbildung auch über formalisiert erworbene Zusatzqualifikationen verfügen und auf diese Weise eine ideale Schnittstelle zwischen den bibliothekarischen Anwender\*innen und den Systemen sind.

Die Betreuung eigener Server wird häufig von übergeordneten Einrichtungen wie universitären, städtischen oder regionalen Rechenzentren oder externen Dienstleistern übernommen. Bei kleinen Bibliotheken werden oftmals auch andere Tätigkeiten von externen Dienstleistern durchgeführt.

Der Umgang mit dem Mangel an IT-Fachkräften wird für die Ressourcenplanung des IT-Managements in Bibliotheken zur Herausforderung werden. Dabei wird auch Open-Source-Software die in der Community entwickelt und unterstützt wird, eine größere Rolle spielen, ebenso wie externe Dienstleister und Software as a Service. Eine umfassende Bedarfsanalyse bei IT-Systemen wird daher zukünftig noch stärker berücksichtigen müssen, wie eine längerfristige Betreuung von eingesetzten Systemen ggf. auch ohne eigenes Personal gewährleistet werden kann.

#### Ressourcenplanung

Veränderungen des Status Quo durch einen Systemwechsel, neue Funktionalitäten oder auch personelle Änderungen durch neue Mitarbeitende, Berentung o.ä. beeinflussen die Personal- und Ressourcenplanung. Eine kontinuierliche Beschäftigung mit diesem Thema ist notwendig, um das [Betriebsrisiko von IT-Systemen](#betriebssicherheit-und-risikomanagement) zu minimieren. Die Benutzung von gut etablierter und weit verbreiteter freier Software verursacht Kosten durch Kompetenzaufbau oder -einkauf, macht unabhängig in Bezug auf Datenhoheit, Datensicherheit und vor allem die Weiterentwicklung. Die Lizenznahme eines kommerziellen Produktes lässt grundsätzlich weniger Individualisierung zu und verlagert die Verantwortung für die Verfügbarkeit und Betriebssicherheit auf den jeweiligen Anbieter.

Für die Planungen muss entsprechend regelmäßig der Bedarf analysiert werden:

* Systeme: Welche Kompetenzen erfordert der Betrieb der Systeme?
* Personal: Wie viel Personal steht mit welchen Kompetenzen zur Verfügung?
* Wie verteilen sich die Kompetenzen auf das vorhandene Personal?
* Wie hoch ist die Übereinstimmung bei vorhandenen und benötigten Kompetenzen?
* Sind Weiterbildungen sind erforderlich?

Das Thema Aus- und Weiterbildung sowie die Personalgewinnung wird im Folgenden ausführlicher betrachtet.

### Aus- und Weiterbildung

In der Einleitung wird Cody Hanson (2015) zitiert: „Most importantly, all library staff must understand that our software is our library, and is everyone’s responsibility.“ Bezogen auf Einarbeitung und Weiterbildung bedeutet das, dass sich Mitarbeitende mit der (Weiter-)Entwicklung von Software ebenfalls weiterbilden und weiterentwickeln. Nur so kann die Verantwortung von allen Mitarbeitenden mit Bezug zur Bibliotheks-IT gemeinsam getragen werden.

Nachfolgend werden aktuelle Beispiele zur Aus- und Weiterbildungen mit Bezug bibliothekarischen IT-Bereich aufgeführt. Nicht betrachtet werden Szenarien wie die Einarbeitung von Anwender\*innen von IT-Systemen bei der Einführung oder dem Wechsel von Systemen.

#### Ausbildungsmöglichkeiten und Zusatzqualifizierung

Historisch gibt es keine formalisierte Ausbildung für die erwähnten Systembibliothekar\*innen. Die notwendigen Kenntnisse werden klassischerweise im Rahmen von „Training on the Job“ erworben.

Allgemeine Ausbildungen und Studiengänge im Bereich IT und Data Science bieten eine gute Grundlage, decken aber bibliotheksspezifische IT-Themen nur unzureichend ab. Stand 2022 gibt es mehrere spezielle Ausbildungsangebote für die Arbeit in der Bibliotheks-IT mit unterschiedlichen Schwerpunkten:

* berufsbegleitende Master-Studiengänge:
  + [Bibliotheksinformatik an der TH Wildau](https://www.th-wildau.de/studieren-weiterbilden/studiengaenge/bibliotheksinformatik-msc-berufsbegleitendes-studium/)
* Vollzeit-Studiengänge:
  + [Masterstudiengang Digitales Datenmanagement an der FH Potsdam und HU Berlin](https://www.fh-potsdam.de/studium-lehre/studiengaenge/digitales-datenmanagement-m)
* Kurse
  + [Zertifikatskurs Data Librarian an der TH Köln](https://www.th-koeln.de/weiterbildung/zertifikatskurs-data-librarian_63393.php)
  + [Zertifikatskurs Forschungsdatenmanagement an der TH Köln](https://www.th-koeln.de/weiterbildung/zertifikatskurs-forschungsdatenmanagement_82048.php)

Diese Zusatzqualifizierungsmöglichkeiten sind eine sehr gute Möglichkeit, um vorhandene Mitarbeiter\*innen systematisch weiterzuentwickeln und die IT-Kenntnisse in der Bibliothek zu verbreiten. Der Erwerb dieser Abschlüsse ist jedoch zeitaufwändig und passt nur für wenige Lebenssituationen.

#### Weiterbildung

Bibliothekarische Ausbildungsstätten sowie Verbundzentralen sind wichtige Akteure bei der Weiterbildung von Bibliothekspersonal und machen teilweise entsprechende punktuelle Weiterbildungsangebote. Dabei handelt es sich in der Regel um ein- oder halbtägige Angebote, die durchaus im Einzelnen Hilfestellung bieten. Für Mitarbeitende mit Bezug zur Bibliotheks-IT sollten ausdrücklich zeitliche und ggf. finanzielle Ressourcen für die Nutzung dieser Angebote bereitgestellt werden.

Auch der Besuch von Konferenzen ist eine wichtige Säule der Weiterbildung. Im Kontext der Bibliotheks-IT hervorzuheben sind hier

* Jahrestagungen der Verbundzentralen
* Bibliothekstage und -konferenzen
* Tagung der [European Library Automation Group](https://elag.org) (ELAG)
* [Code4Lib-Konferenzen](htts://code4lib.org/conference/) in den USA
* [Access Conference](https://accessconference.ca/) in Kanada
* [UKSG Annual Conference](https://www.uksg.org/events/annualconference) in Großbritannien
* Jahrestreffen der [UXLibs](http://uxlib.org/) in Großbritannien

Die wichtigste Rolle bei der Qualifizierung für die Aufgaben im Bereich Bibliotheks-IT dürfte die informelle Weiterbildung spielen. Informelle Weiterbildungsformen sind

* Anwendungstreffen: z.B. jährlich für DSpace, VuFind, Koha, FOLIO, Kitodo, OPUS, …
* Library Carpentries
* Mailinglisten, Foren und andere Kommunikationskanäle
* persönliche Kontakte, Gruppen wie UX Roundtable
* Fachpublikationen: Code4Lib Journal, Weave Journal
* Soziale Medien: Weblogs, Twitter, Discord

#### Personalgewinnung

Die Gewinnung von Personal für die Aufgaben im Bereich der digitalen Dienste ist neben der Aus- und Weiterbildung eine zweite Herausforderung. Die Gehaltsstruktur im öffentlichen Dienst ist für informatisch ausgebildetes Personal nicht unbedingt wettbewerbsfähig, so dass viele ausgewiesene IT-Stellen nur schwer besetzt werden können. Eine unmittelbare Reaktion darauf kann sein, die Vorteile der Beschäftigung im öffentlichen Dienst besser herauszuarbeiten (unkommerzielles Umfeld, gesellschaftliche Relevanz der Tätigkeiten).

Dennoch ist es erwartbar, dass Aufgaben im Bereich Bibliotheks-IT künftig stärker an Verbundzentralen oder externe Dienstleister outgesourct werden müssen.

Die Ausrichtung der bibliothekarischen Studiengänge wird die Bedarfe bei den digitalen Diensten noch stärker berücksichtigen und Studierende mit einem erhöhten Interesse an den Aufgaben in der Bibliotheks-IT rekrutieren müssen.

## Zusammenfassung und Ausblick

Auch nach Auswahl eines Systems ist eine permanente Beobachtung des [Lebenszyklus](#lebenszyklen-von-it-systemen) erforderlich. Es empfiehlt sich immer eine frühzeitige Reaktion auf sich ändernde Anforderungen. Das Wissen um das System als auch um seine Anwendung müssen ebenfalls aktuell gehalten werden, z.B. durch entsprechende [Fortbildungen oder Schulungen](#aus--und-weiterbildung). Sollte sich ein System-Umstieg abzeichnen, sind vor allem die internen Arbeits-Prozesse zu berücksichtigen: das Wissen der Systemanwendenden und -betreuenden ist somit unverzichtbar, denn nur dadurch kann auf eine Ablösung bzw Anpassung des Systems effektiv reagiert werden.

# Anforderungen an die IT-Entwicklung

|  |
| --- |
| Hinweis |
| Die Entwicklung von Software wird im besten Fall an den Bedürfnissen der [Nutzenden](#einbeziehung) und auf die Nutzbarkeit des Produkts ausgerichtet. Es gibt verschiedene Methoden, die entsprechenden Bedarfe und Anforderungen zu ermitteln und sie in die Entwicklung einzubeziehen. Neben klassischen Verfahren der [Bedarfsermittlung](#bedarfsermittlung) gehörgen dazu unter anderem der Einsatz von Personas, Use Cases oder Storyboards. Weitere [Methoden](#methoden) sind zum Beispiel Storyboards, Wireframes oder auch Prototypen. |

Im Kapitel [Management von IT-Systemen](#management) wurde bereits auf Themen wie Barrierefreiheit und software-ergonomische Anforderungen sowie den permanenten Anpassungsbedarf an Systeme im Laufe ihrer Lebenszeit eingegangen.

Betrachtet man sein persönliches Nutzungsverhalten im digitalen Bereich wird klar, dass sich auch die eigenen Präferenzen bezüglich der Nutzung von Apps oder Webseiten ändern. Ursachen dafür sind beispielsweise Veränderungen an Lebens- oder Arbeitskontexten, Erwartungen an die Bedienbarkeit von Systemen oder durch die digitale Transformation beziehungsweise technischen Fortschritt möglich gewordene neue Nutzungsformen von Medien.

IT-Entwicklung sollte sich daher auch an den [Bedürfnissen](#bedarfsermittlung) von [Nutzenden](#einbeziehung) ausrichten.

## Nutzer\*innenorientierten Gestaltung

Nutzer\*innenorientierte Gestaltung heißt, die Bedürfnisse von Nutzenden in den gesamten Entwicklungsprozess einzubeziehen. Das bedeutet, dass deren Bedarfe nicht nur als Quelle von initialen Anforderungen dienen, sondern kontinuierlich in den Entwicklungsprozess einbezogen werden. Hierbei ist es besonders wichtig, die Fähigkeiten und Bedürfnisse der Nutzenden sowie ihre Arbeitskontexte und -aufgaben in den Entwurf von IT-Systemen einzubeziehen. Diese Aspekte finden sich auch in den zugrundeliegenden Definitionen, wie der Usability (siehe Abschnitt [Was beeinflusst den Nutzungseindruck?](#was-beeinflusst-den-nutzungseindruck)) wieder.

Beim nutzer\*innenorientierten Design oder dem User-Centered Design (UCD) handelt es sich nicht im formale Methoden im engeren Sinn, sondern um eine Sammlung i.d.R. empirisch abgesicherter Techniken mit drei Kernideen (Gould und Lewis 1987).

1. Fokussierung auf Nutzer\*innen und deren Aufgaben von Beginn der Entwicklung an
2. deren kontinuierliche Einbeziehung und Auswertung von Nutzer\*innen-Feedback sowie Performance-Messung
3. Nutzung eines iterativen Design-Prozesses

Kling und Leigh Star ergänzen, dass die ganz individuellen Fähigkeiten der Nutzenden in Betracht gezogen werden müssen (Kling und Star 1998), was allein schon aus Gründen der digitalen Teilhabe sinnvoll erscheint.

Generell zielt UCD darauf ab, interaktive Systeme zu entwickeln, welche einfach zu nutzen und nützlich sind. Hierbei wird ein Fokus auf Aspekte wie Effektivität, Effizienz, Benutzerzufriedenheit und Zugänglichkeit gelegt (Deutsches Institut für Normung e. V. (DIN) 2020). Diese Aspekte werden in Abschnitt [Was beeinflusst den Nutzungseindruck?](#was-beeinflusst-den-nutzungseindruck) weiter erläutert.

Das Central Digital and Data Office des Vereinigten Köngreichs fasst die zentral zu bearbeitenden Arbeitspunkte im nutzer\*innenzentrierten Gestaltungsprozess und den Weg dahin prägnant in seinen „Government Design Principles“ zusammen „Government Design Principles. GOV.UK“ (2012):

1. Start with user needs
2. Do less
3. Design with data
4. Do the hard work to make it simple
5. Iterate. Then iterate again
6. This is for everyone
7. Understand context
8. Build digital services, not websites
9. Be consistent, not uniform
10. Make things open: it makes things better

Unter der oben genannten Website des Central Digital and Data Office finden sich auch umfangreiche Hinweise, wie sich die einzelnen Punkte praktisch umsetzen lassen.

### Was beeinflusst den Nutzungseindruck?

Gut bedienbare, interaktive Systeme sollen Zufriedenheit auslösen und zugänglich sein. Die Erreichung dieser Ziele und zentrale Begriffsdefinitionen sind Teil des Arbeits- und Forschungsgebiets der Software-Ergonomie und finden sich in den einschlägigen Normen wie der DIN EN ISO 9241-11 wieder (Deutsches Institut für Normung e. V. (DIN) 2020).

Von zentraler Bedeutung sind dabei zwei Kernbegriffe: die Usability (Gebrauchstauglichkeit) und die User Experience (Nutzer\*innenerfahrung)

|  |
| --- |
| Tipp |
| **Usability** ist das „Ausmaß, in dem ein System, ein Produkt oder eine Dienstleistung durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um festgelegte Ziele effektiv, effizient und zufriedenstellend zu erreichen“ DIN EN ISO 9241-11  **User Experience** bezeichnet die „Wahrnehmungen und Reaktionen einer Person, die aus der tatsächlichen und/oder der erwarteten Benutzung eines Produkts, eines Systems oder einer Dienstleistung resultieren“ DIN ISO 9241-210:2011 |

Bei Usability handelt es sich um eine Eigenschaft eines Systems, die während der konkreten Interaktion mit diesem relevant wird und beispielsweise angibt, inwiefern Hürden bei der Bedienung auftreten ([Abbildung 2.1](#fig-usux)). Zur Vermeidung von Usability-Problemen existieren ein Vielzahl von Heuristiken, die in den einschlägigen Normen skizziert werden bzw. durch Autoren wie Shneiderman in seinen „8 golden rules“ (Shneiderman und Plaisant 2005) oder Nielsen mit seinen „[10 Heuristics](https://www.nngroup.com/articles/ten-usability-heuristics/)“ benannt werden.

Die User Experience hingen bezieht sich auf die Wahrnehmung der Nutzenden sowohl vor, nach und auch während der Interaktion. Sie bezeichnet sozusagen die Positionierung gegenüber einem System und hat damit Auswirkungen darauf, ob Nutzende ein System erneut benutzen werden oder, z. B. aufgrund von schlechter Bedienbarkeit, d. h. schlechter Usability, vor einer zukünftigen Nutzung zurückschrecken. Es reicht folglich nicht aus, einzelne Aspekte einer Nutzer\*innenschnittstelle zu optimieren. Vielmehr muss der gesamte angebotene Service aus Sicht der Nutzenden optimiert werden, damit sich ein positives Nutzungserlebnis einstellt. Diese Optimierung beschränkt sich dabei nicht nur auf die digitalisierten Anteile eines Services sondern bezieht alle Arbeitsschritte, egal ob analog oder digital, mit ein.

|  |
| --- |
| Abbildung 2.1: Zusammenhang zwischen Usability und User Experience |

## Wie beziehen wir unsere Nutzer\*innen ein?

Mit der Einführung einer neuen IT-Lösung werden bestimmte strategische Ziele verfolgt wie die Ablösung eines veralteten Systems, die Einführung einer neuen Dienstleistung und dergleichen. Die konkrete Ausgestaltung dieser strategischen Ziele sollte unter Einbeziehungen der beabsichtigten Nutzenden erfolgen. Die konsequente Bedarfsorientierung sichert die Qualität der Dienste und verhindert, dass eigene Bedürfnisse und Einschätzungen von Expert\*innen die Entwicklung dominieren. Für die Einbeziehung von Nutzer\*innen gibt es verschiedene Methoden, die im Folgenden kurz dargestellt werden sollen.

### Bedarfsermittlung

#### Klassische Methoden zur Bedarfsermittlung

Zu den in Bibliotheken auch jenseits der Entwicklung von digitalen Diensten häufig genutzten Methoden der Bedarfsermittlung gehören qualitative und quantitative Befragungen sowie Beobachtungen. Diese Methoden sind aus der empirischen Sozialforschung entlehnt. Für viele Software-Projekte sind groß angelegte Befragungen zu aufwändig, allerdings ist es empfehlenswert, sich über Studien aus vergleichbaren Projekten zu informieren und daraus nach Möglichkeiten Ableitungen für eigene Zielsetzungen zu entwickeln.

Beobachtungen können sehr flexibel angelegt und geplant werden. Dadurch können valide Ergebnisse mit vertretbarem Aufwand produziert werden und die Studie bei Bedarf gut skaliert werden. Der Fokus bei solchen Studien liegt darauf, Nutzer\*innen in ihrem Arbeitsalltag zu beobachten, um ihre Herangehensweise bei der Lösung von Aufgaben und Problemen zu ermitteln. Übertragen auf digitale Dienste kann das zum Beispiel im Rahmen eines Usability-Tests passieren, in dem eine oder mehrere Personen ein System nutzen. Typischerweise werden während des Tests nicht nur Notizen oder Aufnahmen gesichert, sondern die Tester\*innen nutzen das Think-Aloud-Protokoll. Dabei sollen Nutzende in Echtzeit laut kommentieren, was sie denken, sehen und tun (siehe Abschnitt [Methoden](#methoden)).

Fokusgruppen dagegen sind eine qualitative Methode, in der Vertreter\*innen verschiedener Zielgruppen gemeinsam an einem bestimmten, vorher formulierten Thema arbeiten. Das können sowohl Diskussionen über Anforderungen und Wünsche an ein bestimmtes System sein, als auch die Planung von Einsatzszenarien oder Workflows. Durch die freie Wahl von Themen und Mitgliedern, z. B. Nutzende ohne Vorerfahrungen und/oder Expert\*innen, sind Fokusgruppen ebenfalls eine sehr flexible, breit anwendbare Methode.

#### Bedarfsermittlung mit Personas und Use Cases

Personas sind fiktive Persönlichkeiten, die stellvertretend für einzelne Zielgruppen eines Dienstes entwickelt werden. Die Beschreibungen enthalten vielfältige Informationen über die Persona und laden damit dazu ein, den zu entwickelnden Dienst aus der Perspektive der jeweiligen Persona zu beurteilen, jenseits von abstrakten Anforderungen. Darüber hinaus helfen Personas dabei, Prioritäten zu setzen und die Zielerreichung zu überprüfen. Es empfiehlt sich, für jedes strategische Ziel eine Persona zu erstellen, mindestens drei bis fünf Personas insgesamt.

Abgeleitet von solchen Personas fällt es häufig leicht, konkrete Use Cases für die Interaktion mit einem System zu definieren. Ein Use Case beschreibt dabei eine Reihe von Aktionen, die eine Person in bzw. mit einem System durchführen kann. Das kann beispielsweise in einem Fließtext passieren, in dem ein Szenario beschrieben wird.

|  |
| --- |
| Abbildung 2.2: Aus einem [Vortrag zum Scenario-based Design](https://opus4.kobv.de/opus4-bib-info/frontdoor/index/index/docId/3526) |

Außerdem kann es sich lohnen, solche Use Cases zu visualisieren. Dabei können Start, Ende, mögliche Verzweigungen, alternative Aktionen und mehr mit verschiedenen Formen modelliert werden. Dafür können formalisierte Systeme wie die Unified Modelling Language (UML) zum Einsatz kommen. Sie bietet ein Set verschiedener Formen, um Start, Ende, Verzweigungen, Alternativen und mehr visuell zu beschreiben. Aber auch Skizzen können Nutzungsszenarien bereits verdeutlichen und als Diskussionsgrundlage dienen, z. B. in Form von Storyboards, die in einem eigenen [Unterkapitel zu dieser Methode](#storyboards-als-frühe-methode) noch beschrieben werden.

Use Cases können sowohl als Grundlage für den Entwicklungsprozess dienen als auch für die Evaluation eines Systems (siehe Abschnitt [Evaluierung](#evaluierung)). Für die Nutzenden-Personas einer Bibliothek kann eine breite Palette von Use Cases existieren. Manche sind dabei eher allgemein zu verstehen, andere bibliotheksspezifisch und natürlich sind alle je nach Einrichtung bzw. Anforderungen beliebig erweiterbar. Zu beachten ist, dass sowohl Personas als auch Use Cases zwingend auf der Grundlage vertrauenswürdiger Daten wie denen aus der Bedarfsermittlung erstellt werden sollten. Solche Methoden ohne Kenntnisse der Zielgruppen anzuwenden kann nur zur Reproduktion der eigenen Meinung führen.

### Methoden

Testaufgaben für Usability-Tests werden erstellt, um typische Nutzungsszenarien mit Hinblick auf die Usability des Systems hin zu überprüfen. Die folgenden Methoden können relativ einfach umgesetzt werden, generieren jedoch bereits wertvolle Erkenntnisse.

#### Think-Aloud-Protokolle

Die zentrale Idee bei Think-Aloud-Protokollen ist, dass Proband\*innen während der Interaktion mit dem zu evaluierenden System ihre Meinungen, Gedanken und Gefühle laut aussprechen.

Dadurch wird es den Beobachter\*innen ermöglicht, zuvor unsichtbare, kognitive Prozesse der Proband\*innen zu beobachten sowie einen Einblick in typische Nutzungsweisen zu gewinnen. Durch die Verbalisierung und Beschreibung des Systems durch die Nutzenden lernt man zeitgleich die Nutzer\*innenterminologie für bestimmte Sachverhalte kennen, die teils erheblich von der Fachsprache abweichen wird. Die Ergebnisse der Methode können z. B. durch Notizen oder Audioaufnahmen festgehalten werden.

#### Co-Discovery Learning

Die Kernherausforderung bei der Erstellung von Think-Aloud-Protokollen ist es, die Proband\*innen kontinuierlich zu motivieren, selbst kleinste Gedanken zu verbalisieren. Beim Co-Discovery Learning arbeiten zwei Testpersonen gleichzeitig an einem System und helfen sich gegenseitig bei der Erfüllung der Aufgaben. Dadurch entstehen Gespräche und gewissermaßen automatisch ein Think-Aloud-Protokoll beider Personen.

Die Methode bildet einerseits eine realistische Arbeitssituation des gegenseitigen Helfens ab und normalisiert andererseits das laute Aussprechen von Gedanken innerhalb einer Dialogsituation.

#### Quantitative Methoden

Beobachtungsmethoden generieren primär qualitative Daten, ebenso wie viele Inspektionsmethoden. Aus Managementsicht werden jedoch oft Entscheidungen auf Grundlage von quantitativen Daten bevorzugt, da diese häufiger als Fakten wahrgenommen werden.

Einfache, relativ leicht zu erhebende quantitative Metriken im Rahmen von Usability-Tests sind z.B.:

* Nutzungsfehler pro Zeiteinheit,
* Anzahl nicht benötigter Befehle (Menus, Icons, Links)
* Benötigte Zeit für den Abschluss einer Arbeitsaufgabe (insbesondere im Vergleich mit einer vorherigen Iteration)
* Benötigte Anzahl an Klicks/Links, um an ein bestimmtes Ziel zu kommen.

Der „Benutzungsfragebogen ISONORM 9241/10“ bietet einen interessanten Kompromiss zwischen qualitativen und quantitativen Daten, da er qualitative Aussagen bezüglich der Usability eines Systems (z.B. Aufgabenangemessenheit und Selbstbeschreibungsfähigkeit) mithilfe einer siebenstufigen Likert-Skala abbildet. Der [Fragebogen](https://people.f3.htw-berlin.de/Professoren/Pruemper/instrumente.html) ist frei im Internet verfügbar. Beachtet werden muss, dass für belastbare quantitative Daten die Größe der Testgruppe deutlich steigen muss, um Verfälschungen durch Einzelpersonen zu vermeiden.

### Einbeziehung von Nutzenden in die Entwicklung

Als Grundlage für Personas oder Use Cases und alle weiteren Schritte ist die Einbeziehung von tatsächlichen Nutzenden in die Entwicklung also bereits in einem frühen Stadium möglich und sinnvoll. Diese Einbeziehung sichert ab, dass wesentliche Ziele der Nutzenden erreicht werden und in mitunter komplexen Entwicklungsprozessen die richtigen Schwerpunkte gesetzt werden. Dafür stehen verschiedene Methoden zur Verfügung.

Nachfolgend werden drei Ansätze vorgestellt:

* Storyboards - Skizzierung von Interaktionskonzepten
* Wireframes und Mock-Ups - Skizzen der Oberflächen
* Prototypen - erste funktionsfähige Iterationen

#### Storyboards als frühe Methode

Ein Storyboard illustriert, wie ein User Interface (UI, Nutzer\*innenoberfläche) auf Eingaben reagiert ohne das Interface visuell perfekt darzustellen. Es kann genutzt werden, um in Use Cases bestimmte Aktionen zu illustrieren.

|  |
| --- |
| hier sollte man am besten etwas mit Bib-Bezug kritzeln, das bild ist nur als Platzhalter zu verstehen z.B. <https://www.storyboardthat.com/de/storyboards/1c78733f/matilda-library-visit> |

Die Visualisierung von Interaktionsideen kann Beteiligten helfen, mögliche Abläufe nachzuvollziehen. Storyboards sind dabei oft leichter verständlich als z. B. technische Diagramme mit der oben genannten UML. Trotzdem ist darauf zu achten, dass Ideen und Konzepte für Stakeholder und Nutzende klar beschrieben werden, um Missverständnisse zu vermeiden. IN dieser Form lassen sich Storyboards nutzen, um z.B. verbale Beschreibungen oder Nutzungsszenarien zu ergänzen.

Durch die noch vage Darstellung der Idee können dann Diskussionen angeregt werden. Beispielsweise können Storyboards in Fokusgruppen vorgestellt und diskutiert oder auch in Einzelgesprächen mit verschiedenen Stakeholdern analysiert werden. Möglichst alle Fragen und Ideen sollten dabei ohne Limitierungen behandelt werden können und die Ergebnisse festgehalten werden.

Vor- und Nachteile von Storyboards im Überblick:

| Vorteile | Nachteile |
| --- | --- |
| leicht verständlich, für alle Stakeholder geeignet | nicht jeder Use Case oder jede Interaktionsmöglichkeit ist darstellbar |
| bereits im frühen Entwurfsprozess einsetzbar | digitale, nichtlineare Produkte (z.B. Websites) sind schwer darstellbar |
| schnelle Erstellung ohne Vorkenntnisse möglich | ggf. Unklarheiten bei der Nutzung (z.B. durch unklare Symbole) |

#### Wireframes und Mock-Ups

Wireframes und Mock-Ups werden vor allem dazu genutzt, erste Skizzen für Struktur, Layout und Funktionalitäten eines Interface vorzustellen. Ähnlich wie Storyboards dienen sie als einfach zu erstellende Diskussionsgrundlage, mit deren Hilfe ein Abgleich der Vorstellungen von einem System und der Gestaltungsmöglichkeiten durchgeführt werden kann.

|  |
| --- |
| Abbildung 2.3: <https://www.mockplus.com/blog/post/basic-uiux-design-concept-difference-between-wireframe-prototype> (Platzhalter) |

Ein Wireframe („Drahtmodell“) ist eine noch undetaillierte („low-level“) Ausarbeitung eines Interfaces, v. a. darauf ausgerichtet, die Positionierung der einzelnen Elemente zu planen. Daher sind z.B. Bilder oder Buttons als Kästchen dargestellt, Text als Striche und ähnliches (siehe [Abbildung 2.3](#fig-5)). Ein Mock-Up ist, im Kontext Design, eine ausgereifte („high-level“) Version des Interfaces mit realistischen Farben, Schriftarten und Elementen. Sowohl Wireframes als auch Mock-Ups sind also rein statische Entwürfe des zukünftigen Produkts im Gegensatz zu Prototypen, die interaktiv sind und damit echte Funktionalitäten enthalten.

#### (Interaktive) Prototypen

Die nächsthöhere Form eines geplanten Produkts, (interaktive) Prototypen, besitzen bereits erste Funktionen des geplanten Interfaces. Auch hier gibt es eine Spanne von rudimentären, low-level bis hin zu ausgereiften, high-level Prototypen, die durch Iterationen schrittweise erreicht werden. Üblich ist außerdem die Unterteilung in „vertical slice“, die qualitativ hochwertige Umsetzung nur eines bestimmten Teils des Produkts, und „horizontal slice“, die prototypische Umsetzung einer möglichst großen Bandbreite des späteren Systems.

Erste Prototypen müssen dabei noch nicht zwingend programmiert werden, sondern können durch entsprechende Prototyping Software, wie [Figma](https://www.figma.com) oder [Axure](https://www.axure.com), umgesetzt werden. Diese besitzen eine Art Bausystem für Interfaces mit mehreren Ansichten, die über Aktionen wie den Klick auf einen Button verbunden werden können. So kann Nutzenden gewissermaßen ein Produkt vorgetäuscht werden, das dann mit rudimentären Funktionen bereits getestet werden kann.

Während des eigentlichen Softwareentwicklungsprozesses wird der anfängliche Prototyp mit jeder Iteration hochwertiger und nimmt mehr den Charakter eines vollen Systems an. Es empfiehlt sich, nach Iterationen regelmäßig zu evaluieren, ob neue Funktionen oder Änderungen noch für die Zielgruppen geeignet sind.

### Evaluierung

Die vorangegangenen Abschnitte haben herausgestellt, wie wichtig es ist, regelmäßig Feedback der Nutzenden zu erhalten. Eine zentrale Datenquelle dafür ist die Begleitung eines Projekts durch Evaluierungen. Ein Beispiel für eine lebendige Evaluierungskultur ist das “[User Research Center” der Harvard Library](https://urc.library.harvard.edu/), das regelmäßig verschiedene Methoden anwendet, um Angebote gemeinsam mit Nutzenden zu evaluieren und diese öffentlich in einem [Wiki](https://wiki.harvard.edu/confluence/pages/viewpage.action?pageId=232199222) teilt.

Im Rahmen der Usability-Evaluierung entscheidet man dabei grob zwei Methoden: Beobachtungs- und Inspektionstests ([Abbildung 2.4](#fig-6)). Während erstgenannte unter Einbeziehung von Nutzer\*innen durchgeführt werden, werden Inspektionstests häufig durch Usability-Expert\*innen realisiert.

|  |
| --- |
| Abbildung 2.4: |

Als Vorteil der Beobachtungstests erweist sich aus der Praxissicht, dass diese auch ohne eine formale Usability-Ausbildung durch engagierte Mitarbeiter\*innen durchgeführt werden können. Im Folgenden soll deshalb das prinzipielle Vorgehen bei einem Beobachtungstest skizziert werden.

#### Testgruppen

Die Testgruppe muss die potentielle Nutzungsgruppe bestmöglich repräsentieren, jedoch nicht sehr groß sein. Die Erfahrung zeigt, dass ca. fünf Testpersonen ausreichen, um die wichtigsten Usabilityprobleme eines Systems zu identifizieren Jakob Nielsen (2000). Statt eines einzigen Tests mit vielen Teilnehmenden bieten sich daher schnell durchzuführende Tests mit wenigen Teilnehmenden an, um ein Produkt iterativ zu verbessern. Möchte man jedoch verschiedene Typen von Nutzer\*innen analysieren oder quantitative Ergebnisse sammeln, muss die Gruppengröße entsprechend wachsen.

Neben den typischen Streuungsmerkmalen wie demographischen und kulturellen Faktoren (z.B. Bildungshintergrund) bietet es sich an, Nutzer\*innen auszuwählen, die über ein unterschiedliches Maß an Vorwissen über das zu entwickelnde oder verwandte Produkte verfügen. Außerdem sollten Personen integriert werden, welche von Einschränkungen betroffen sind, die in Abschnitt [Accessibility](#accessibility) thematisiert wurden.

#### Testablauf und Vorbereitungen

Nach der Rekrutierung repräsentativer Nutzer\*innen und der Vorbereitung der benötigten Materialien und der Testumgebung bietet sich ein Pilottest mit Proband\*innen an. Dieser dient der Validierung der eigenen Annahmen über die Testaufgaben (s.u.) und die Machbarkeit des Ablaufs.

Die Testumgebung sollte eine entspannte und natürliche Arbeitsumgebung vermitteln. Diese ist in jedem Fall einer künstlichen Laborumgebung vorzuziehen. Während der Beobachtungstests ist sicherzustellen, dass keine Unterbrechungen, z.B. in Form von Telefonanrufen erfolgen, damit die Proband\*innen das zu evaluierende System konzentriert testen können.

Nach dem Beobachtungstest sollte es den Proband\*innen ermöglicht werden, die Testergebnisse zu erhalten. Außerdem ist es neben dem obligatorischen Dank für die Teilnahme üblich, eine Aufmerksamkeit - je nach Dauer z. B. Kaffee, Süßes, Gutscheine - auszuhändigen, um die eigene Wertschätzung für das zeitliche Investment der Proband\*innen auszudrücken. In einer Erklärung zum Datenschutz ist die anonyme Datennutzung zuzusichern.

#### Testaufgaben

Wie die Testgruppen müssen auch die Testaufgaben repräsentativ für den späteren Einsatzzweck des Systems sein. Die von den Proband\*innen zu bearbeitenden Testaufgaben müssen realistische Aufgaben in bzw. mit dem System sein und in der gegebenen Zeit absolvierbar sein. Dabei ist zu beachten, dass sich die Arbeitsaufgaben an tatsächlichen Use Cases orientieren und nicht trivial sind.

Die Formulierung der Arbeitsaufgaben muss unmissverständlich für die Proband\*innen sein und auf deren (mitunter variierendes) Vorwissen eingehen. Ein Pilottest hilft, dies zu überprüfen.

Die Gestaltung der einzelnen Aufgaben sollte einer Dramaturgie folgen, um die Proband\*innen während des gesamten Tests zu motivieren. Das heißt konkret, dass die ersten Teilaufgaben leicht zu lösen sein sollten und deren Schwierigkeit dann kontinuierlich zunimmt, um durch komplexere Aufgaben belastbare Aussagen zu erhalten.

## Zusammenfassung und Ausblick

Es gibt verschiedenste Methoden mit denen Bedarfe ermittelt und Nutzende in die Entwicklung von Software einbezogen werden können - je nach Umfang des Produkts und des Anwender\*innenkreises. Usertests erfordern ein anderes Zeitmanagement als die Entwicklung von Personas. Auch der Anwendungsfall nimmt Einfluss auf die Methodenauswahl. So kann für die Entwicklung eines neuen Designs die Verwendung von Wireframes und Mockups bei der Bedarfsermittlung hilfreich sein. Wird ein Portal mit neuen Interaktionsmöglichkeiten entwickelt, empfehlen sich Prototypen, mit denen auch die Interaktionen getestet werden können.

# Technische Infrastruktur

|  |
| --- |
| Hinweis |
| Diese Hauptkapitel wird erst noch erstellt (Siehe [entsprechendes Issue](https://github.com/pro4bib/handbuch-it-in-bibliotheken/issues/41)) |

# Bibliotheksmanagementsysteme

|  |
| --- |
| Hinweis |
| Bibliotheksmanagementsysteme sind IT-Systeme, die die [Kernprozesse](#prozessabbildung) einer Bibliothek unterstützen. Das modular aufgebaute System verfügt über verschiedene [Komponenten](#komponenten) für die jeweiligen Kernprozesse Erwerbung, Katalogisierung/Erschließung und Ausleihe sowie für die Recherche. Die [Entwicklungsgeschichte](#geschichte) der Managementsysteme in Bibliotheken erstreckt sich mittlerweile über drei Generationen. Relevant für die Arbeit mit den Managementsystemen ist auch die [Anbindung an Verbundkataloge](#verbundkataloge) sowie die [Integration anderer Systeme](#integration). |

In diesem Text wird der Begriff **Bibliotheksmanagementsystem (BMS)** verwendet. Damit ist ein Softwareprodukt gemeint, mit dem die Arbeitsprozesse rund um die Erwerbung, Erschließung, Ausleihe, Zugriff und Auffindbarmachung von Bibliotheksbeständen über Kataloge abgebildet und automatisiert werden können. Teilweise wird im Deutschen auch der allgemeinere Begriff *Bibliothekssystem* verwendet. In der angloamerikanischen Literatur finden sich die Begriffe *Integrated Library System (ILS)* und *Library Management System (LMS)*, zuletzt aber auch *Library Services Platforms*.

Das BMS spielt eine zentrale Rolle für die meisten klassischen Geschäftsprozesse in Bibliotheken. Durch die Ausweitung der Aufgaben in den Bereichen Publikationsdienste, Open Science oder auch Lernort sind in neuerer Zeit jedoch noch weitere Aufgaben hinzugekommen, die durch die klassischen BMS nicht abgebildet werden. Darüber hinaus haben die frühen Systeme nur sehr unzureichende Möglichkeiten, die nötigen Informationen zu elektronischen Ressourcen und ihrer Zugänglichkeit abzubilden. Auch zur Unterstützung von neueren Aufgaben wie der Publikationsunterstützung oder der Verwaltung räumlicher Ressourcen werden separate Systeme genutzt. Daraus ergibt sich der Bedarf, das BMS an diese separaten Systeme anzubinden, was die Bedeutung von Schnittstellen und offenen Architekturen erhöht hat.

## Übersicht

### Geschichte

|  |
| --- |
| Hinweis |
| Die erste Generation der Bibliotheksmanagementsysteme umfasste Grundfunktionen für die Ausleihe wie Verbuchung und Mahnung, für die Erwerbung die Verwaltung von Bestellungen und teilweise auch über über Telnet erreichbare Kataloge für die Bibliotheksnutzenden. Die zweite BMS-Generation verfügt über erweiterte Funktionalitäten zur Unterstützung der Kernprozesse sowie durch Weboberflächen der Kataloge aus. Die dritter Generation zeichnet sich durch stärkere Modularisierung und mehr Schnittstellen zur Anbindung weiterer Systeme aus. |

Bibliotheksmanagementsysteme unterstützen alle Kernprozesse des bibliothekarischen Arbeitens, das heißt die Erwerbung, Erschließung und Ausleihe von Medien. Ihren Ursprung haben Bibliotheksmanagementsystem in den 1960er Jahren, als Bibliotheken damit begannen, Katalogdaten untereinander auszutauschen und auf diese Weise Prozesse zu optimieren. In dieser Zeit entwickelten sich auch die heute noch gebräuchlichen Austauschformate für Katalogdaten, z.B. MARC.

In den 1970er Jahren erlaubte die fortschreitende technische Entwicklung, die Automatisierung weiterer Prozesse über den Datenaustausch hinaus. Zunächst war dies vor allem die Ausleihe mit der Verbuchung von Medien und der Erzeugung von Mahnschreiben. Auch die Verwaltung von Bestellungen im Rahmen der Erwerbung wurde möglich, so dass man in der Folge von *Integrated Library Systems* zu sprechen begann. Davon, dass Katalog-, Erwerbungs- und Nutzer\*innen-Daten an einem Ort gehalten und bearbeitet wurden, versprach man sich eine größere Effizienz der Arbeitsprozesse. Diese **erste Generation** von BMS beinhaltete teilweise auch schon digitale Funktionen für Bibliotheksnutzer\*innen wie über Telnet erreichbare Kataloge, die von Anfang an als integraler Bestandteil der BMS gesehen wurden (Borgman 1997).

Die Entstehung des World Wide Web in den 1990er Jahren hatte zunächst vor allem Einfluss auf die Benutzbarkeit der Kataloge, die Web-Oberflächen erhielten. Aber auch die anderen Komponenten der BMS wurden überarbeitet, und zwar zunehmend auch von kommerziellen Anbietern, während die ersten Systeme als Eigenentwicklungen von Bibliotheken entstanden. Die Landschaft an **Systemen der 2. Generation** war von den späten 1990er bis in die Nullerjahre sehr divers, ist zuletzt aber von vielen Übernahmen geprägt worden, so dass man von einem konsolidierten Markt sprechen kann (Breeding o. J.).

Seit den 2010er-Jahren vermarkten Anbieter eine neue Generation von BMS, die **Next-Generation Library Management Systems**, die auch **Library Services Platforms** genannt werden. Diese zeichnen sich durch verschiedene technische und funktionale Neuerungen aus. Die Datenhaltung erfolgt in der Regel cloudbasiert (auch wenn dies bei bei älteren Systemen im Grunde auch schon der Fall war), außerdem werden in der Regel mehr Schnittstellen zur Integration des Systems mit anderen Lösungen angeboten. Funktional wurden die Systeme vor allem um die Möglichkeit der Verwaltung von elektronischen Ressourcen erweitert sowie Statistik- und Reporting-Funktionalitäten verbessert.

Seit dem Ende der 1990er Jahre spielen auch wieder Lösungen eine Rolle, die nicht kommerziell sind. Diese Open Source-Lösungen haben in der Regel eine große Anwender Community und lassen einen vielfältigen Markt für Support- und Wartungsdienstleistungen zu.

|  |
| --- |
| Evolution der Bibliotheksmanagementsysteme |
| Quelle: Matthews und Block (2020), S. 7 [TODO: Ins Deutsche adaptieren]  Nach Matthews und Block (2020) lässt sich die Geschichte der BMS in sechs überlappende Epochen einteilen:   1. **System-Epoche**: Erste Schritte in den 1950er bis in die 1970er-Jahre hin zur Entwicklung von Software, z. T sehr experimentell, die die klassischen Geschäftsgänge von Bibliotheken in einem digitalen System abbilden sollten - dadurch prägte sich der Begriff "Bibliothekssystem". Das Augenmerk bei der Entwicklung lag besonders auf Nachbildungen des Leihverkehrs unter besonderer Beachtung der Identifikation überfälliger Medien 2. **Epoche der Funktionalität**: Kommerzielle Bibliothekssoftware-Anbieter begannen sich zu formieren, die erstmals eine integrierte Lösung der verschiedenen Automationsbereiche (Erwerbung, Katalogisierung, Zeitschriftenakzession, Verbuchung, Leihverkehr usw.) anboten. Hierdurch entstand die Bezeichnung "Integriertes Bibliothekssystem" (IBS), der auf den aus dem US-amerikanischen Raum übernommenen Begriff "Integrated Library System" (ILS) zurückgeht. In den 1980ern entstanden die ersten Online-Kataloge (OPAC), die die in Bibliotheken traditionellen Zettelkataloge nachbildeten. 3. **Nutzer\*innen-Fokus-Epoche**: Durch die Erkenntnis, dass sich die Gewohnheiten von Bibliotheks-Nutzer\*innen im Zugang zu und Umgang mit Medien u.a. mit dem Aufkommen des WWW in ihrem Alltag zunehmend ändern (z.B. durch die Nutzung von Online-Shopping und Suchmaschinen), rückten die Bedürfnisse der Nutzer\*innen immer mehr in den Fokus bei der Entwicklung von Bibliothekssystemen 4. **Epoche der Verbreiterung der Informationsressourcen**: Der Übergang in eine Phase, bei der Medien nicht mehr erworben, sondern digital lizenziert werden. Entsprechend entwickelt sich das Bedürfnis nach einem Electronic Resource Management (ERM) und neuartige BMS unterfüttern zum Ende der Epoche diesen Wandel mit einer von Medientypen unabhängigen Ressourcenverwaltung. 5. **Discovery-System-Epoche**: Systeme, die über den lokalen Medienbestand hinaus auch extern lizenzierte Inhalte über eine alleinige Suchplattform zugänglich machen, erfreuen sich zunehmender Beliebtheit bei den BMS-Betreiber\*innen. Sie sollen den Nutzer\*innen einen deutlichen Mehrwert bieten. Seit den 2010er Jahren sind es überwiegend kommerzielle Verlage, die umfangreiche e-Medien-Pakete oder Indizes von Volltext- und Bibliografie-Datenbank als lizenzierbare Resource Discovery Services anbieten. 6. **Wissensinnovation**: Bibliotheken realisieren überwiegend, dass Discovery-Systeme nicht ihre gewünschte Wirkung entfalten und sie sich deutlicher von Plattformen großer Tech-Unternehmen abgrenzen müssen. Wissen soll neu erschlossen werden mit innovativen Technologien wie 3D-Druck, Virtual Reality (VR), Open-Access-Repositorien etc. |

### Komponenten

BMS sind in der Regel modular aufgebaut und verfügen mindestens über Module für folgende Funktionen:

* Erwerbung
* Katalogisierung / Erschließung
* Ausleihe
* ein Recherche-Modul, das sich vorwiegend an die Bibliotheksnutzenden richtet

Die Systeme der **1. und 2. Generation** können als sehr ausgereift bezeichnet werden und lassen vielfältige Möglichkeiten zu, bibliothekarische Geschäftsgänge in einem hohen Detaillierungsgrad abzubilden. Nachfolgend werden diese entsprechenden Aufgabenbereiche skizziert.

#### Erwerbung

meint die Beschaffung benötigter Bestände bei Verlagen. Darunter fallen z.B. folgende Aufgabengebiete:

* Bestellungen
* Budgetverwaltung
* Rechnungsverwaltung
* Lieferantenverwaltung
* Zeitschriften- und Fortsetzungsabonnements
* Unterstützung [EDIFACT](https://unece.org/trade/uncefact/introducing-unedifact)-Standards
* Buchbinder

*Siehe auch* [*Prozessabbildung: Erwerbung*](#erwerbung-prozess)

#### Katalogisierung

meint die Erschließung der verwalteten Medien und digitalen Quellen, z.B. anhand

* Übernahme von Fremddaten
* Anbindung an Verbünde
* Integration digitalisierter Medien

*Siehe auch* [*Prozessabbildung: Katalogisierung*](#katalogisierung-prozess)

#### Ausleihe

meint vorwiegend die Verwaltung physischer Medien bzw Objekte, regelt Interaktionen mit Nutzer\*innen wie z.B.:

* Abbildung komplexer Reglements nach Benutzer- und Medientypen, Standorten usw (siehe auch [Benutzungsbedingungen](#benutzungsbedingungen))
* Ausleihfristen
* Verwaltung von Standorten
* Versand von Benachrichtigungen
* Anbindung an Selbstverbuchungslösungen
* Mahngebühren

*Siehe auch* [*Prozessabbildung: Ausleihe*](#ausleihe-prozess)

#### Recherchemodul

Sicht für die Nutzer\*innen auf Bestände der Einrichtung zur Recherche und Kontofunktionen

* Katalog (auch OPAC genannt)
* Benutzerkonto

*Siehe auch* [*Prozessabbildung: Katalog*](#katalog-prozess)

Die **Next Generation-Systeme** zeichnen sich gegenüber den Systemen der 1. und 2. Generation in der Regel durch andere Systemarchitekturen aus. Das heißt, sie verfügen über aktuellere technische Einzelkomponenten und Schnittstellen, auf deren Grundlage auch zahlreiche zusätzliche Funktionalitäten angeboten werden können. Im Einzelnen gibt es folgende Merkmale, die ein **Next Generation-System** kennzeichnen (Schweitzer 2016):

* Angebot als Software as a Service ([SaaS](#cloud))
* Mandantenfähigkeit
* Interoperabilität durch offene, standardisierte und dokumentierte Schnittstellen
* Verfügbarkeit von Datenbanken bzw. Knowledge Bases für bibliografische Daten und Lizenzinformationen
* oftmals kein fest integrierter Katalog, sondern Schnittstellen zu Discovery-Systemen
* Statistik-Werkzeuge
* Erzeugung von Semesterapparaten
* Anlegen von Favoriten-Listen
* Anzeige von Buchcovern

### Aktuell relevante Software-Produkte

Stand Mitte 2022 haben folgende aktuellen BMS die derzeit größte Marktreife und -durchdringung im deutschsprachigen Raum (Sortierung nach Name):

|  | Organisation | Marktstatus | Open Source | Individuelle Entwicklung |
| --- | --- | --- | --- | --- |
| aDIS/BMS | aStec | ÖBs und WBs vor allem im BSZ | nein | durch aStec |
| Alma | ExLibris | WBs in Berlin, NRW, Bayerische Staatsbibliothek, Schweiz | nein | durch ExLibis, integrierte Apps in Eigenregie |
| Folio | Open Library Foundation | Einführung in WBs | ja | in Eigenregie oder durch Dienstleister |
| Koha | Koha Community | ÖBs und Spezialbibliotheken, in Planung im KOBV | ja | in Eigenregie oder durch Dienstleister |
| LBS | OCLC | WBs im GBV und Spezialbibliotheken | nein | durch VZG |
| LIBERO | LIBERO/Knosys | ÖBs und WBs | nein | durch LIBERO |

Weitere BMS wie ExLibris Aleph, SISIS Sunrise und allegro werden zwar auch noch an vielen Bibliotheken eingesetzt, aber nicht mehr wesentlich weiterentwickelt. Das Cloud-basierte System WMS von OCLC ist in Deutschland bislang nur vereinzelt im Einsatz. Für BibliothecaPlus ist von OCLC ein Nachfolger angekündigt.

Darüber hinaus gibt es mehrere kommerzielle Systeme, deren Funktionsumfang auf bestimmte Arten von Bibliotheken zugeschnitten ist, beispielsweise:

* [Perpustakaan](https://must.de/default.html?Lib.htm) ist in Schulbibliotheken verbreitet und wendet sich auch an nicht-bibliothekarisch vorgebildetes Personal
* [NOS](https://de.wikipedia.org/wiki/NOS_(Bibliothekssoftware)) ist in internen Forschungs- und Behörden-Bibliotheken verbreitet
* Quria von [Axiell](https://www.axiell.com/) ist in skandinavischen ÖBs verbreitet und löst im deutschsprachigen Raum das BMS BIBDIA ab

Eine umfangreiche internationale Übersicht von BMS enthält der von Marshall Breeding gepflegte [Library Technology Guide](https://librarytechnology.org/products/). Für den Deutschsprachigen Raum gibt es Übersichten von Verbundzentralen oder Büchereifachstellen, z.B. Kluge (2022) für öffentliche Bibliotheken. Darüber hinaus sind Daten zu BMS systematisch in Wikidata erfasst und können beispielsweise unter <https://w.wiki/574K> abgefragt werden.

|  |
| --- |
| Tipp |
| Der IT-Lebenszyklus von BMS ist mit mehr als 20 Jahren im Vergleich zu anderen IT-Systemen eher lang. So wurde beispielsweise Folio im Rahmen des *Open Library Environment Project* bereits 2009 initiiert und wird wahrscheinlich erst im nächsten Jahrzehnt in die Wartungsphase übergehen. |

### Vergleich mit anderen Managementsystemen

Aufgrund der hohen Kosten für die Einführung oder die Migration eines BMS dürfte sich für viele Entscheider\*innen die Frage stellen, ob sich die Investition lohnt bzw. ob sich die Aufgaben auch mit anderen Lösungen erledigen lassen (siehe hierzu [Beschaffung und Marktanalyse](#beschaffung-und-marktanalyse)).

Systeme zur Automatisierung von Geschäftsprozessen gibt es in verschiedenen Branchen. Eine genauere Betrachtung der Aufgaben, die durch Automatisierung unterstützt werden sollen, kann aufzeigen, ob dafür ein Bibliotheksmanagementsystem oder eine andere Lösung besser geeignet ist.

Die folgenden Alternativen sind möglicherweise für kleine Einrichtungen relevant, die über sehr überschaubare Bestände verfügen und kaum oder wenig ausleihen:

Erfassung von Medien:

* Listen in einer Tabellenkalkulation (Excel, LibreOffice, …)

Erfassung und Web-Präsentation von Medien:

* [Library Thing for Libraries](https://www.librarything.com/forlibraries)
* [Zotero Groups](https://www.zotero.org/groups/)
* Stand-Alone-Lösungen für Electronic Resource Management wie [Coral](http://coral-erm.org)

Ausleihe

* Plugins für Wordpress wie [WebLibrarian](https://www.greengeeks.com/tutorials/create-a-library-management-system-in-wordpress/)

Erwerbung

* Finanzbuchhaltungssysteme wie SAP, [HIS-Hochschul-ERP](https://www.his.de/)

Nutzerdatenverwaltung

* IDM-Systeme

Bibliotheken mit einem jährlichen Zuwachs von über 500 Medien und verschiedenen Nutzertypen und Ausleihbedingungen werden ist die Nutzung eines BMS zu empfehlen, da hier eine gewisse Prozesseffizienz einerseits und eine Erschließungs- und Dienstleistungsqualität andererseits zu erreicht werden können. Als gedankliches Experiment ist die Überlegung, auf ein BMS zu verzichten, jedoch gut geeignet, um sich über die Anforderungen klar zu werden. Insbesondere die Rolle des Bibliothekskataloges als Schnittstelle zu den Bibliotheksnutzer\*innen kann und sollte kritisch hinterfragt werden. Beispielsweise gab es [Überlegungen der Universitätsbibliothek in Utrecht](https://insights.uksg.org/articles/10.1629/2048-7754.174/), auf dieses klassische Instrument gänzlich zu verzichten.

### Integration des BMS mit anderen IT-Systemen

Innerhalb der Bibliothek werden BMS meist zusammen mit anderen Softwaresystemen eingesetzt. Insbesondere sind dies:

* [Selbstbedienungsautomaten](#automatisierung-und-selbstbedienung) (Ausleihe, Rücknahme, Sortierung von Medien, Bezahlung von Gebühren)
* Dokumentenserver, Content Management Systeme und andere Repositorien
* Workflowsysteme (Digitalisierung von Altbestand; Publikationsunterstützung, …)

Weitere Systeme müssen für eine effektive Arbeit sinnvoll mit dem BMS verbunden werden:

* Haushaltssysteme wie SAP, HIS Haushalt-ERP
* [Identitätsmanagementsysteme](#identity-management) (Account-Verwaltung)
* Lieferantensysteme (bibliographische Daten, Bestell- und Rechnungsdaten)

Im bibliothekarischen Umfeld sind folgende Systeme relevant:

* der [Verbundkatalog](#verbundkataloge)
* die [Zeitschriftendatenbank](#Xef3cdf849515896b72e4da871885dcc60b5fbed) (siehe unten)
* die [elektronische Zeitschriftendatenbank](#Xef3cdf849515896b72e4da871885dcc60b5fbed) (siehe unten)

Für die regionale und überregionale Literaturversorgung (physische, Print-Medien, E-Medien) spielt die Anbindung an folgende Systeme eine wesentliche Rolle

* Fernleihe
* Dokumentenlieferdienste (wie [Subito](https://www.subito-doc.de/) und Fachinformations-Lieferdienste)

Die Anbindung an die entsprechenden Dienste (Zentraler Fernleih-Server, Fernleihdienst, Subito-Server etc.) ist für viele, aber durchaus nicht alle Bibliotheken relevant.

Im Zusammenhang mit dem Aufbau der Fachinformationsdienste für die spezialisierte Informationsversorgung in Deutschland werden in zunehmendem Maße Fachportale entwickelt. Relevante Katalog-Informationen werden aus möglichst vielen Bibliotheken regelmäßig abgerufen (Harvesting), in ein einheitliches Datenformat übertragen und anschließend als gemeinsamer Index für die übergreifende Recherche in Discovery-Systemen angeboten. Die BMS müssen entsprechend über [Standardschnittstellen](#datenformate-und-schnittstellen) die relevanten Katalogdaten in einem vereinbarten [Datenformat](#bibliographische-metadaten) bereitstellen.

### Verbundkataloge

In Deutschland haben sich Katalogverbünde in den 1970er und 1980er Jahren entwickelt. Zunächst haben sich die wissenschaftlichen Bibliotheken meistens auf Bundesland-Ebene für die Rationalisierung der Katalogisierung zu Verbünden zusammengeschlossen. Inzwischen sind in diesen Verbünden auch öffentliche Bibliotheken vertreten. Darüber hinaus gibt es mit WorldCat einen internationalen Verbundkatalog. Die Anbindung an WorldCat geschieht jedoch in der Regel nicht direkt über das lokale BMS sondern über den Bibliotheksverbund.

Folgende Bibliotheksverbünde gibt es:

| Verbund | Verbundkatalog | System |
| --- | --- | --- |
| BVB | B3Kat | ALEPH (Ex Libris) |
| BSZ | K10plus | CBS (OCLC) |
| GBV | K10plus | CBS (OCLC) |
| hebis | hebis | CBS (OCLC) |
| hbz | hbz | Aleph (Ex Libris)  Alma-Netzwerkzone (Ex Libris) |
| KOBV | B3Kat | Aleph (Ex Libris) |
| VÖBB (öffentliche Bibliotheken) | VÖBB | aDIS/BMS (aStec) |
| Österreichischer Bibliothekenverbund | OBV | Alma-Netzwerkzone (Ex Libris) |
| Swiss Library Service Platform (SLSP) | swisscovery | Alma |

### Anbindung an Verbundkataloge/Verbundkatalogisierung

Die Übernahme von bibliografische Daten oder - bei elektronischen Medien - Paket -bzw. Lizenzinformationen aus anderen Systemen ist für eine Bibliothek unabhängig davon, ob sie in einem Verbund organisiert ist, von Interesse. Eine Anbindung von bibliografischen Datenquellen, z.B. per Z39.50, für die Übernahme der entsprechenden Daten gilt daher als Mindeststandard. In Verbünden organisierte Bibliotheken katalogisieren in der Regel bereits in Verbunddatenbanken und wollen die Katalogisate dann verzögerungsfrei in die lokalen Systeme übernehmen.

Für Informationen zu elektronischen Medien gibt es neben den Verbunddatenbanken weitere Datenbanken bzw. Knowledge Bases, aus denen Paket- und Lizenzinformationen hervorgehen. Diese sind zum Beispiel

* die Zeitschriftendatenbank (ZDB) als zentrales Nachweissystem für Zeitschriften und Fortsetzungen in deutschen und österreichischen Bibliotheken
* die GoKB als kooperativ gepflegte Knowledge Base für elektronische Ressourcen

### Statistik und Reporting

Mitunter verfügen BMS über eigene Module für die Erstellung von Statistiken. Folgende Statistiken sind typischerweise erforderlich:

* Arbeitsstatistiken - Für die tägliche Arbeitsorganisation und die Abrechnung der Arbeitsleistungen gegenüber den Stakeholdern müssen in regel- und unregelmäßigen Abständen Statistiken und Bericht aus dem BMS erstellt werden. Die Inhalte werden von den Stakeholdern bestimmt.
* Deutsche Bibliotheksstatistik - Bibliotheken können sich entscheiden, Daten für die [Deutsche Bibliotheksstatistik](https://www.bibliotheksstatistik.de/) zu erfassen. Die notwendigen Daten sollten über das BMS ermittelt werden können. Durch die einheitliche Definition der statistischen Kennzahlen ist eine umfassende, vergleichende Auswertung aller Bibliothekssparten (wissenschaftliche, öffentliche, Spezialbibliotheken) möglich.
* Sonderstatistiken wie Statistiken der Fachinformationsdienste (FID)

Bei den Systemen der 1. und 2. Generation ist es bisweilen nötig, zusätzliche Werkzeuge zum Einsatz zu bringen, um alle gewünschten Berichte zu erstellen (z.B. *BibControl* oder *Crystal Reports*). Während die integrierten Module vor allem auf die Daten des eigenen Systems fokussiert sind, können externe Werkzeuge auch Fremddaten aufnehmen, zum Beispiel Daten aus Besucherzählern.

## Bibliotheksorganisation

Bei der Implementierung oder Anpassung eines BMS ist die Organisation der Bibliothek, die Gestaltung der Prozesse sowie die räumliche Situation zu berücksichtigen. Handelt es sich zum Beispiel um einen öffentliche oder wissenschaftliche Bibliothek? Ist die Organisation als ein einschichtiges oder zweischichtiges System angelegt? Ist es eine einzelne Bibliothek oder eine Zentralbibliothek mit Zweigstellen?

Auch die Aufstellung der Medien innerhalb der Gebäude nimmt Einfluss auf die Ablauforganisation und damit die Konfiguration des Systems. Dies lässt sich anhand der folgenden Beispiele darstellen:

* Bücher können in einer anderen Zweigstellen ausgeliehen werden.
* Die Bibliothek verfügt über einen Magazinbestand, also physische Medien, die für die Nutzenden nicht unmittelbar zur Verfügung stehen.

In beiden Fällen muss auch der Bestellprozess über das System abgebildet werden. Im Magazin bzw. der Zweigstelle sind der Anschluss und die Aufstellung von Druckern für die Erzeugung von Bestellzetteln zu berücksichten. Sind die Medien für die Bibliotheksnutzenden direkt zugänglich, entfällt der Bestellschritt und der abzubildende Prozess beginnt mit der Ausleihverbuchung.

Auch das Rechtemanagement eines BMS ist abhängig von der Größe der Organisation. So sind ggf. verschiedene Berechtigungsstufen für die Bearbeitung von Daten im BMS für die Bibliotheksbeschäftigen einzuführen. Die Berechtigungen bilden die Arbeitsorganisation ab und berechtigen z.B. zum Lesen, Anlegen, Editieren oder Löschen von Ausleihbestellungen, Benutzenden- oder Katalogdaten, Erwerbungsunterlagen, Gebühreninformationen u.ä.

### Benutzungsbedingungen

Die Benutzungsbedingungen werden durch die Ausleihpolitik der Bibliothek bestimmt. Die Gestaltung der Bedingungen erfolgt sowohl bezogen auf die Medien und die Bibliotheksbenutzenden. Dabei geht es um die Frage, was von wem ausgeliehen werden darf und, wenn eine Ausleihe möglich ist, wie und für welchen Zeitraum diese erfolgen kann.

Eine Grundlage zur Abbildung der Benutzungsbedingungen ist die Definition von Benutzungsgruppen. Die Benutzungsgruppen werden durch verschiedene Kriterien charakterisiert. Zur Illustration zwei Beispiele:

* Gruppenbildung Universitätsbibliothek:
  + intern: Studierende, Lehrende, weitere Universitätsangehörige
  + extern: externe Wissenschaftler\*innen, interessierte Öffentlichkeit
* Gruppenbildung öffentlichen Bibliothek: Kinder, Jugendliche, Erwachsene

Die Einteilung von Bibliotheksnutzenden in Gruppen dient der einfacheren Zuweisung von Rechten und Ausleihbedingungen, wird aber auch für statistische Zwecke genutzt. Die Ausleihpolitik bestimmt, welche Rechte den verschiedenen Benutzungsgruppen zugewiesen werden. So erfolgt z.B. die Gruppeneinteilung in öffentlichen Bibliotheken i.d.R. nach dem Alter. Einerseits wird damit die Zugänglichkeit der Medien für Kinder und Jugendliche gesteuert. Andererseits dient diese Gruppierung der Einstufung der Gebühren (Kinder und Jugendliche zahlen oft weniger oder keine Gebühren).

Neben den Gruppen werden Ausleihbedingungen auch auf die Medien bezogen. Zur Illustration:

* Präsenzbestände vs. ausleihbare Medien,
* besonders wertvolle Medien oder
* elektronische Publikationen, die nur unter bestimmten Bedingungen und von bestimmten Benutzergruppen genutzt werden können.

Benutzungsbedingungen werden also sowohl durch die Zugehörigkeit zu einer Benutzungsgruppe als auch durch das Medium selbst bestimmt. Die Beschreibung der Benutzungsbedingungen ist somit eine wesentliche Voraussetzung für die Einrichtung des Ausleihmoduls eines BMS.

## Datenverwaltung in BMS

Ein BMS verwaltet zum einen Daten über die von der Bibliothek bereitgestellten oder vermittelten Ressourcen (vor allem physische und digitale Medien) und zum andern Daten über wesentlichen Arbeitsprozesse (beispielsweise Erwerbung und Ausleihe). Dabei lassen sich grob drei Arten von Daten unterscheiden:

* **Bibliographische Metadaten** zur Beschreibung von Ressourcen
* **Digitale Inhalte** wie Volltexte, Digitalisate und andere digitale Publikationen
* **Verwaltungsdaten** zur Unterstützung von Workflows

Die Datenhaltung erfolgt in der Regel in relationalen Datenbanken (MySQL, Oracle).

Zur sinnvollen Verarbeitung von Daten im BMS und in Integration mit anderen System müssen Daten bestimmten Datenformaten entsprechen, über Schnittstellen abruf- und ggf. änderbar sein und Mindestanforderungen an die Datenqualität genügen.

### Bibliographische Metadaten

Bibliographische Metadaten in Form von Titel-, Exemplar- und Normdaten bilden den Kern den klassischen Katalogs. An anderer Stelle tauchen diese Daten beispielsweise als Suchindizes für Discovery-Systeme auf. Das BMS verwaltet diese Daten um Ressourcen Auffindbar und Zugreifbar zu machen. Diese Daten können von verschiedenen Bibliotheken gemeinsam genutzt und in der Regel frei zur Verfügung gestellt werden (siehe Kapitel zu Open Data).

### Digitale Inhalte

Dies sind letztendlich die Daten die für die Nutzenden der Bibliothek vor allem von Interesse sind. Im Falle von Open Access Publikationen bietet das BMS nur einen möglichen Weg zum Zugriff, für erworbene oder lizenzierte Inhalte muss das BMS dagegen unterschiedliche Zugriffsrechte unterstützen.

Digitale Inhalte werden in der Regel nicht direkt im BMS sondern in eigenen Content Management Systemen (CMS) und Repositorien verwaltet. Ein BMS muss mit diesen Systemen durch Verwendung gemeinsamer Datenformate, Import, Export und Verlinkung zusammenarbeiten können. Der Unterschied zwischen Metadaten und Inhalten ist dabei mitunter fließend und hängt vom Anwendungsfall ab. Reicht es oft Publikationen grob mit Metadaten zu beschreiben, so umfasst in anderen Fällen die Erschließung von Publikationen auch Dokumentstrukturen und inhaltliche Bestandteile wie z.B. einzelne Abbildungen.

### Verwaltungsdaten

Verwaltungsdaten dienen der Unterstützung von Arbeitsabläufen innerhalb der Bibliothek (siehe [Prozessabbildung](#prozessabbildung)). Diese Daten sind zum größten Teil nicht öffentlich und müssen insbesondere im Falle von Daten von Nutzenden im Rahmen des [Datenschutz](#datenschutz) vertraulich behandelt werden.

Zur Interoperabilität mit anderen Informationssystemen innerhalb der eigenen oder übergeordneten Einrichtung gibt es in der Regel nur wenig übergreifend etablierte Standards und Schnittstellen, so dass hier oft zusätzliche Anpassungen an das BMS notwendig sind.

### Datenformate und Schnittstellen

Da Computer nicht selbständig mitdenken und interpretieren können, müssen Daten nach klar definierten Regeln aufgebaut sein. Diese Regeln sollten möglichst genau dokumentiert sein. Damit verschiedene Systeme Daten austauschen können, sollten möglichst etablierte Standardformate verwendet werden.

Trotz gemeinsamer Standards ist ein genaues Hinschauen immer erforderlich, da sich die Handhabung gleicher Formate in der Praxis zwischen verschiedenen Systemen und Einrichtungen unterscheidet.

Neben Standardformaten gibt es speziellere Anwendungsformate. Diese basieren allerdings in der Regel auf allgemeinen Strukturierungssprachen (*CSV*, *XML*, *JSON* oder *RDF*) die je nach BMS besser oder schlechter unterstützt werden.

* Beispiele für bibliographische Standardformate sind *MARC21*, *BIBFRAME* und als kleinster gemeinsamer Nenner Dublin Core. Das PICA-Format bzw. darauf aufbauende Formate ist vor allem als Internformat in den Bibliotheksverbünden GBV, BSZ und an der DNB verbeitet.
* Verbreitete Metadaten-Schnittstellen sind Z39.50, *SRU* und *OAI-PMH*.
* Beispiele für relevante Formate und Schnittstellen für digitale Inhalte sind PDF, *METS/MODS* und *IIIF*.
* Beispiele für relevante Schnittstellen für BMS-Verwaltungsdaten sind *LDAP* und *PAIA*.

Eine umfassende Übersicht von Datenformaten mit Schwerpunkt auf Formate, die für Bibliotheken relevant sind, bietet die Seite <https://format.gbv.de>.

### Datenqualität

Im Gegensatz zu physischen Dingen ist Daten von außen nicht anzusehen, ob sie unvollständig, veraltet oder aus anderen Gründen fehlerhaft sind. Ohne kontrolliertes Qualitätsmanagement muss davon ausgegangen werden, dass die Qualität von Daten kontinuierlich abnimmt. Zur Ermittlung und Verbesserung der Datenqualität tragen bei:

* **Richtlinien** legen einheitliche Regeln für Daten fest, beispielsweise durch Katalogisierungsregeln wie RDA (Soll-Stand)
* **Validierung** ermittelt die Übereinstimmung von Daten mit formal definierten Vorgaben (Ist-Stand)
* **Statistiken** geben quantitative Auskunft, zum Beispiel über die Anzahl erfolgreich importierter oder exportierter Datensätze

Nicht zuletzt beeinflussen auch die Möglichkeiten der Ein- und Ausgabe von Daten ihre Qualität, beispielsweise über die Usability der Katalogisierung.

## Marktanalyse und Beschaffung

|  |
| --- |
| Tipp |
| Der deutschsprachige [BMS-Markt 2022](#aktuelle-sw-produkte) ist überschaubar. Für den [Entscheidungsprozess](#entscheidungsprozess) sind daher vor allem auch der Umfang der gewünschten und gewichteten Funktionalitäten, Varianten des Betriebs (gehostet oder lokal) oder auch die Mitgliedschaft in einem Verbund als Kriterien heranzuzuziehen. |

Die Beschaffung eines BMS ist für eine Bibliothek eine große Herausforderung, nicht nur wegen der zu kalkulierenden Kosten sondern auch wegen des erheblichen Einflusses auf alle bibliothekarischen Arbeitsschritte. Der Aufwand für die Migration von Altdaten, die Revision von Geschäftsgängen und die Schulung von Personal muss bei der Beschaffung berücksichtigt werden. Nicht zuletzt ist die Wahl eines BMS auch eine strategische Entscheidung, da die Möglichkeiten auf zukünftige Anforderungen einzugehen je nach System und eigenen Ressourcen unterschiedlich ausfällt.

Es kann auch eine ethisch-moralische Entscheidung oder ein Commitment zu einer ökologisch-nachhaltigen Betriebsführung (öffentliche Einrichtungen als Vorzeigecharakter für einen ökologischen Wandel) sein, die Aspekte der nachhaltigen Beschaffung zu berücksichtigen, wie sie sich bei BMS als auch anderen IT-Anwendungen stellen, etwa die Konsequenzen des ökologischen Fußabdrucks der genutzten Infrastruktur (z. B. CO2-Ausstoß des Rechenzentrums).

Auch aus datenschutzrechtlicher Perspektive gibt es Voraussetzungen zu berücksichtigen, die gegen die Anschaffung bestimmter BMS-Lösungen sprechen (siehe Abschnitt [Datenschutz](#datenschutz)).

Vor diesem Hintergrund ist die Auswahlentscheidung für einzelne Bibliotheken oft ein langwieriger Prozess. Bei Teilnahme an einem Verbund können sich Bibliotheken durch diesen über die BMS, die vom Verbund unterstützt werden, informieren und beraten lassen (siehe [Übersicht deutscher Verbundsysteme](https://www.gbv.de/bibliotheken/geografische-uebersicht-der-deutschen-verbundsysteme)). Die Beschaffung und Einführung von BMS liegt immer in der Verantwortung der jeweiligen Bibliothek oder der Einrichtung, zu der die Bibliothek gehört.

Die Gründe für einen Systemwechsel sind primär technischer oder finanzieller Natur. Beispielhaft werden folgend einige Gründe aufgezählt:

* Das Altsysteme ist technisch überholt oder wird nicht mehr gewartet.
* Es fehlen Schnittstellen für die Integration des BMS in die lokale Informationsinfrastruktur.
* Die Kosten für den laufenden Betrieb sind zu hoch und sollen mit einem anderen System gesenkt werden.
* Eine Funktionserweiterung, z.B. für die Verwaltung von elektronischen Ressourcen, kann an dem bestehenden System nicht mehr vorgenommen werden.

### Entscheidungsprozess

Ein Entscheidungsprozess umfasst typischerweise folgende Schritte:

1. **Workflowanalyse:** Dokumentation bestehender und zukünftig gewünschter Prozesse, die mit dem BMS abgebildet werden sollen
2. **Anforderungsanalyse:** Zusammenstellung und Priorisierung der gewünschten Funktionalitäten und strategischen Zielen unter Einbeziehung aller Stakeholder
3. **Marktanalyse:** Auswahl der in Frage kommenden Lösungen und Betriebsmodelle
4. **Evaluation:** Vertiefte Beschäftigung mit einer Auswahl von Lösungen durch Ausprobieren von Test-Installationen und Kontakt mit Anwendungsbibliotheken
5. **Aufwandsabschätzung** von Migration, Einrichtung und Schulung
6. **Ausschreibung**, falls erforderlich
7. **Auswahlentscheidung**

### Auswahlkriterien eines BMS

Es kann davon ausgegangen werden, dass die aktuell am Markt verfügbaren Systeme die klassischen Geschäftsgänge (siehe Kapitel [Prozessabbildung](#prozessabbildung)) einer Bibliothek gut abbilden können. Die Anforderungen aus dem Kapitel Nutzer\*innenzentrierte Gestaltung gelten grundsätzlich natürlich auch hier.

Die Betrachtung einzelner Systeme einschließlich der Nutzungsszenarien und Use Cases kann sehr aufwändig werden. Daher empfiehlt es sich, die gewünschten Funktionalitäten zu bestimmen und durch die Stakeholder bewerten zu lassen. Die Bewertung kann beispielsweise in Form einer Matrix geschehen, in der die Funktionalitäten nach ihrer Bedeutung/Wichtigkeit einerseits und den zu erwarteten Aufwänden andererseits eingeordnet werden.

|  |
| --- |
| Abbildung 4.1: Beispiel für eine Matrix zur Einordnung von Funktionalitäten |

Zur Evaluierung der BMS können bestehende Anforderungskataloge für die Evaluierung von BMS herangezogen werden, zum Beispiel der gemeinsam [von HBZ und VZG entwickelte Kriterienkatalog](https://www.folio-bib.org/?page_id=247). Dieses umfangreiche Dokument zeigt die Anforderungen an alle Komponenten auf Grundlage der Analyse von sehr ausgereiften Prozessen in Altsystemen auf. Es empfiehlt sich, insbesondere diejenigen Funktionalitäten genau zu überprüfen, die strategisch von besonderer Bedeutung sind.

### Marktanalyse

Da es sich bei BMS um relativ spezialisierte Software handelt und in den letzten Jahren einige Produkte aufgekauft oder eingestellt wurden, ist der Markt sehr überschaubar (siehe Kapitel [aktuell relevanten BMS](#aktuell-relevante-software-produkte)).

Neben der Wahl konkreter Produkte gibt es grundsätzlich drei Möglichkeiten:

* Beitritt zu einem Bibliotheksverbund und Nutzung eines BMS, das von diesem Verbund unterstützt wird, zu den jeweils gültigen Konditionen
* Lizenzierung eines kommerziellen BMS und Einkauf einschlägiger Dienstleistungen für Hosting, Wartung und Support sowie Migration und individuelle Konfiguration
* Implementierung und individuelle Konfiguration eines Open Source-BMS, entweder in Eigenregie oder durch vollständige oder punktuelle Unterstützung von einschlägigen Dienstleistern für Hosting, Wartung und Support sowie Migration und individuelle Konfiguration

Die Vor- und Nachteile im Überblick:

|  | Verbund | kommerzielles BMS | Open Source BMS |
| --- | --- | --- | --- |
| Vorteile | regelmäßige Produktentwicklung  gewisser State-of-the-Art garantiert  klare Kosten- und Leistungsstruktur  große Anwendungscommunity | regelmäßige Produktentwicklung  einheitlicher Leistungsumfang  klare Verantwortlichkeiten | niedrige Anschaffungskosten  große Anwendungscommunities  oftmals regelmäßige Produktenwicklung  viele Dienstleister, die Services rund um Migration, Betrieb und individuelle Anpassung anbieten  offene Schnittstellen und Formate |
| Nachteile | begrenzte individuelle Anpassung  Wartezeiten bei individueller Anpassung | eher geringe individuelle Anpassbarkeit  relativ hohe und intransparente Preise  Abhängigkeit bei der Weiterentwicklung  evtl. Verlust der Datenhoheit  z.T. proprietäre (herstellerspezifische) Systeme und Schnittstellen | erfordert eigene IT-Kapazitäten oder Outsourcing  Risiko der Sicherung von Nachhaltigkeit und Kompatibilität |

Verbünde bieten in der Regel ein oder zwei Lösungen an, die entweder kommerziell oder Open Source sind. Die Mitgliedschaft in Verbünden kann ein kostengünstiger Weg sein, um mit einem BMS und dazugehörigen Dienstleistungen versorgt zu werden. Allerdings steht möglicherweise nicht allen Bibliotheken die Mitgliedschaft in einem Verbund offen oder bedingt andere Nachteile (z.B. den Zwang, an der Fernleihe teilzunehmen und begrenzte Möglichkeiten zur individuellen Anpassung der Software).

Bei kommerzieller Software fallen typischerweise Lizenzkosten an, die sich nach der Größe der Bibliothek oder der übergeordneten Einrichtung richten (z.B. an der Anzahl von Mitarbeitenden, Studierenden oder Einwohner). Dabei werden einmalige Beschaffungs- und jährlichen Wartungskosten unterschieden. Es muß klar vereinbart werden, welche Dienste mit den Wartungskosten (Support, Update auf neue Versionen, …) abgegolten sind.

Der Betrieb der Lösungen kann von den Anbietern oder anderen Dienstleistern (Verbund, andere kommerzielle Anbieter) übernommen werden (Cloud/Software as a Service), d.h. die Bibliotheken brauchen keine eigenen Server zur Verfügung stellen und administrieren. Eine Installation auf eigenen Servern (On-Premise-Lösung) erfordert hingegen eigenes, ausgebildetes Personal.

Bei Open Source-Lösungen gibt es keine initialen Anschaffungskosten. Bei Verfügbarkeit entsprechender Server-Infrastruktur und erfahrenem Personal kann eine Bibliothek die Software selbst installieren und in Betrieb nehmen oder diese Leistungen von Dienstleistern einkaufen.

Die initiale Konfiguration sowie die Migration von Daten aus einem Alt-System können ebenso von den Bibliotheken selbst durchgeführt werden oder sind Teil des Kauf-/Wartungsvertrages.

Die laufende Betreuung des Betriebs von BMS erfordert speziell geschultes und berechtigtes Personal - sogenannte System-Bibliothekar\*innen. In wenigen Fällen wird die Systembetreuung an Dienstleister (beim Hoster) übergeben.

## Prozessabbildung

Für den Einsatz eines BMS bilden Prozessbeschreibungen bzw. Workflows eine wesentliche Grundlage. Auf der Basis der Abbildung der Kernprozesse wie Erwerbung, Katalogisierung, Ausleihe sowie der Rolle des Systems und anderer Akteure können Anpassungen (leichter) vorgenommen werden.

Zur Modellierung, Dokumentation und Visualisierung von Workflows bietet sich klassischerweise eine Modellierungssprache wie BPMN (Business Process Model and Notation) an. Für diese und verwandte Sprachen existieren umfangreiche Werkzeuge und Toolchains, mit denen einerseits Prozesse erstellt werden können, gleichzeitig aber auch – sollte das nötig sein – die modellierten Prozesse automatisiert werden können. Im Endeffekt bedeutet dies, dass aus dem Prozessmodell Programmcode erzeugt wird.

|  |
| --- |
| [Quelle: Stkl (CC-BY-SA 3.0)](https://commons.wikimedia.org/wiki/File:BPMN-1.svg)  Beispiel einer BPMN-Prozessabbildung |

Lässt man die Aspekte der Prozessautomatisierung oder Codegenerierung außer acht, so lässt sich auch eine abgespeckte BPMN-ähnliche Semantik nutzen, um Prozesse zu dokumentieren und zu visualisieren. Andere Alternativen zur Modellierung finden sich in den verschiedenen Diagrammformen der UML (Unified Modelling Language).

Aus heutiger Sicht sollten für die im Folgenden genannten Bereich Prozessbeschreibungen erstellt werden, damit potentielle BMS an Hand dieser geprüft werden können. Hierbei könnten sich Notwendigkeiten für Änderungen in den Prozessabläufen der Bibliothek ergeben, die auf Basis der Beschreibungen genauer adressiert werden können.

### Nutzende

Als Nutzende werden in diesem Kapitel diejenigen Menschen bezeichnet, die mit einem BMS interagieren. Man unterscheidet zwischen den Bibliotheksbeschäftigten, die mit dem Modulen Ausleihe, Erwerbung, Katalogisierung, ERM etc. interagieren, und den Bibliotheksnutzenden (oft auch als Leser\*innen bezeichnet), die mit dem BMS über das Modul OPAC oder nur indirekt über ein Discovery-System oder ein anderes Drittsystem mit dem BMS in Kontakt kommen.

### User-Interfaces für verschiedene BMS-Anwender\*innen

Die Bibliotheksbeschäftigten und die Bibliotheksnutzenden haben verschiedene Sichtweisen auf ein BMS. Bibliotheksbeschäftigte müssen über das User-Interface bei ihrer Arbeit spezifisch durch die Workflows geführt werden. Dabei ist auf eine einheitliche Benutzungsführung und Gestaltung der Oberfläche zu achten.

Für die Bibliotheksnutzenden steht die Information über die Dienste der Bibliothek, deren Bestand und die Nutzung des Bestandes im Vordergrund. Bibliotheksnutzende kommen dabei häufig mit mehreren IT-Systemen in Kontakt (BMS- OPAC-Modul, Web-Server, Discovery-System, …). Daher sollte auch hier auf eine einheitliche Oberfläche der eingesetzten IT-Systeme geachtet werden, auch bezüglich Accounts und Login, zumindest aber auf ein einheitliches Design und eine einheitliche Benutzerführung.

Es ergeben sich daraus die folgenden Anforderungen

1. Intuitive Benutzbarkeit
2. Barrierearme Gestaltung
3. Responsivität

Diese Themen werden auch in den Abschnitten zu [rechtlichen Rahmenbedingungen](#rechtliche-rahmenbedingungen) und im Kapitel zu den [Anforderungen an Bibliotheks-IT](#anforderungen) angesprochen.

### Erwerbung

Ein BMS sollte das Bibliothekspersonal bei den folgenden Aufgaben unterstützen:

1. Überprüfen von vorhandenen Beständen (Vorakzession)
2. Aufgabe von Bestellungen bei definierten Lieferanten auf verschiedenen Wegen
3. Verwaltung von Lieferantendaten
4. Anlegen und Verwalten von Bestellungen von Zeitschriften und Fortsetzungswerken
5. Überwachung von Bestellungen
6. Anlegen und Verwalten von Budgets
7. Akzessionierung von Medien
8. Rechnungsverwaltung inkl. Schnittstellen für haushalterische Systeme
9. Verwaltung von Bindeaufträgen
10. Verwaltung von Nicht-Kauf-Bestellungen

Diese Aufgaben lassen sich mit den am Markt befindlichen Systemen in der Regel gut abbilden. Allerdings werden die meisten Bibliotheken für die Verwaltung von notwendigen Bestellungen von Materialien jenseits des Bibliotheksbestandes (Büromaterial, IT-Ausstattung etc.) zusätzliche haushalterische Systeme einsetzen. Das Erwerbungsmodul ist insofern meist nur eine Komponente im Haushaltswesen.

### Verwaltung von elektronischen Ressourcen

Für die Verwaltung elektronischer Ressourcen sollten folgende Aufgaben unterstützt werden:

1. Erfassung von Lizenzinformationen nach unterschiedlichen Erwerbungsmodellen wie Pakete, Allianz- oder Nationallizenzen
2. Zuordnung von digitalen Inhalten zu Paketen
3. Verwaltung von Paketen
4. Bezug von bibliografischen Daten von Aggregatoren und Verlagen
5. Unterstützung der direkten Verlinkung auf Volltexte aus Katalogen und Discovery-Systemen
6. Auslieferung von aussagekräftigen Zugangsinformationen in Kataloge und Discovery-Systeme
7. Unterstützung bei der Bereitstellung von digitalen Inhalten jenseits von proprietären Apps

Die BMS der 1. und 2. Generation haben erhebliche Defizite bei der Verwaltung von elektronischen Ressourcen. Die Bereitstellung von entsprechenden Funktionalitäten ist daher ein Alleinstellungsmerkmal von BMS der neuen Generation.

Alternativ können aber auch separate, sogenannte Electronic Resource Management-Tools eingesetzt werden (Coral, GoKB und LAS:eR).

### Katalogisierung

Bei der Katalogisierung müssen folgende Tätigkeiten unterstützt werden:

Erfassung von unterschiedlichen Medientypen gemäß aktueller Metadaten-Standards

1. Übernahme von Katalogdaten aus Bibliotheksverbünden
2. Möglichkeit der Integration von Normdaten
3. Erfassung von lokalen Daten
4. Konfigurierbarkeit von Erfassungsmasken

### Katalog

Der Katalog ist die Sicht für die Bibliotheksnutzer\*innen auf die Bestände der Bibliothek. An das Katalogmodul werden folgende Anforderungen gestellt:

1. Web-Interface nach aktuellen Standards bezüglich Barrierefreiheit, Responsivität etc.
2. Angebot von Möglichkeiten der Suche nach bekannten Titeln
3. Angebot von Möglichkeiten der Suche nach Themen
4. Filterung von Trefferlisten nach formalen oder inhaltlichen Kriterien bzw. Standorten
5. Anzeige von Verfügbarkeitsinformationen
6. Anzeige von Neuerwerbungslisten
7. Kontobezogene Funktionalitäten (Einsicht, Verlängerung, Vormerkung, Bestellungen)
8. Anzeige von Neuigkeiten und wichtigen Links auf der Startseite des Katalogs
9. Anpassbarkeit der Katalogoberfläche an das Corporate Design (wenigstens Logo und Farbschema)

In der Geschichte der BMS war das Katalogmodul eher ein Nebenprodukt der Katalogisierungsarbeit. Durch die Veränderungen im Informationsverhalten seit Entwicklung des WWW ist insbesondere auf das Katalogmodul ein besonderer Innovationsdruck entstanden. Auf diesen Druck haben Bibliotheken mit dem Angebot von Discovery-Systemen reagiert, die als alternative Benutzungsschnittstelle zu den klassischen OPACs aufgebaut wurden und neben einem modernen Design auch Suchmaschinen-typische Funktionen wie Facettierung oder Unterstützung bei der Formulierung von Suchbegriffen bieten. Diese Funktionen sind in den BMS der neuen Generation standardmäßig enthalten.

Es entstanden durch den erwähnten Innovationsdruck verschiedene Konstrukte, die Daten der Bibliothek den Nutzenden zur Verfügung zu stellen:

1. Klassischer Katalog (OPAC) als Bestandteil des BMS
2. Katalog als separates Modul (nicht Bestandteil des BMS), selbst entwickelt, zugekauft oder als Open Source
3. Discoverysystem als Bestandteil des BMS: Daten aus dem eigenen Bestand sowie Fremddaten, die als Metadaten zur Verfügung stehen.
4. Discoverysystem als zugekauftes Modul eines anderen Herstellers oder als Eigenbau mit zugekauften Metadaten oder als Open Source mit offenen Daten oder zugekauften Metadaten.

Bei Punkt 4 entsteht die Herausforderung, die im BMS gehosteten Informationen, zum Beispiel über den Ausleihstatus/Verfügbarkeit, auch in der Oberfläche des Discovery-Systems aktuell darzustellen.

### Ausleihe

Ein BMS sollte die folgenden Aufgaben der Ausleihe unterstützen:

1. Anlegen von Nutzergruppen, Standorten, Medienarten
2. Abbildung der in den Benutzungsordnungen festgelegten Ausleihbedingungen, z.B. Leihfristen nach Nutzergruppen, Standorten, Medienarten
3. Verbuchung von Medien (Ausleihe, Rücknahme)
4. Konfiguration von Ausdrucken für Bestellzettel und Vormerkungen
5. Ermöglichen von Bestellungen und Vormerkungen
6. Mahnwesen (Fristen, Mahnstufen)
7. Benachrichtigungen für Bestellungen, Vormerkungen, Mahnungen, Leihfristerinnerungen
8. Gebührenverwaltung
9. Erzeugung von Listen (überfällige Medien, nicht abgeholte Vormerkungen)
10. Anbindung an Bezahlsysteme (Kassenautomaten, Online-Bezahlsysteme)
11. Anbindung von Verfügbarkeits- und Kontoinformationen an Discovery-Systeme
12. Anbindung an Automatisierungslösungen und externe Verbuchungssysteme (etwa mittels RFID)

Die Parametrisierung der Ausleihe ist ein besonders komplexer Bereich der BMS-Installation aufgrund der Vielzahl von zu beachtenden Benutzungsregeln, der Sensibilität der Daten und der besonderen Relevanz eines reibungslosen Betriebs beim Versand von Benachrichtigungen. Ein Beispiel für eine solche Komplexität ist die das Verhalten bei Feiertagen: Hier muss ein Schließtagekalender regelmäßig gepflegt werden, um zu vermeiden, dass Leihfristenden auf Feiertage oder Wochenenden fallen.

### Automatisierung und Selbstbedienung

Als Automatisierung wird die Möglichkeit bezeichnet, die Geschäftsgänge einer Bibliothek mit digitalen Werkzeugen abzubilden und durchführen zu können. Dazu sind Maschinen notwendig, die die entsprechenden Funktionen anbieten. Das schon recht betagte Standardprotokoll für die Kommunikation zwischen BMS und Automat ist [SIP2](https://en.wikipedia.org/wiki/Standard_Interchange_Protocol). Dieses Protokoll hat den Nachteil, dass es ohne Verschlüsselung entwickelt wurde und daher - sofern es sich um ein BMS in der Cloud handelt zumindest - über [stunnel](https://en.wikipedia.org/wiki/Stunnel) verschlüsselt getunnelt wird. Moderne BMS unterstützen mittlerweile zusätzlich auch allgemeine Kommunikationsprotokolle, etwa über [REST](https://de.wikipedia.org/wiki/Representational_State_Transfer), sodass das Tunneln von Verbindungen nicht mehr nötig ist. Außerdem ist man nicht mehr daran gebunden, dass anzubindende Geräte SIP2 unterstützen, was deutlich mehr Marktalternativen öffnet.

Nachfolgend werden Automaten für die Selbstbedienung im Bereich der Ausleihe dargestellt.

#### Selbstverbucher / Ausleihautomaten

Selbstverbucher / Ausleihautomaten bestehen meist aus einer Auflagefläche für die auszuleihenden Medien, einer Schnittstelle für Bibliotheksausweise sowie einem PC, der die Endgeräte verwaltet und mit dem BMS kommuniziert. Bei einer funkgestützten Medienerkennung (RFID) gibt es die Möglichkeit der Stapelverbuchung, es werden also vom Automaten mehrere gestapelte Bücher erkannt und zur Verbuchung angeboten. Bei einer Barcodegestützten Medienerkennung wird jedes Medium einzeln verbucht.

Bibliotheksausweise gibt es in verschiedensten Ausprägungen: Barcode (1D-Code), Funkchip (u.U. proprietär, Bsp.: Intercard), QR-Code (2D-Code). Die 1D- oder 2D-Codes können entweder auf Papier oder in einer App auf dem Smartphone beigebracht werden. Die Schnittstelle im Automaten muss auf die vorhandenen Ausweistypen vorbereitet sein.

Bei Nichtvorhandensein einer separaten Rückgabeanlage kann der Selbstverbucher / Ausleihautomat auch eine Rückgabefunktion anbieten. Zumeist sammeln die Benutzenden die zurückgegebenen Medien unsortiert; im Anschluss erfolgt die Sortierung durch das Bibliothekspersonal.

Nach der Rückgabe- oder Ausleihverbuchung muss der Selbstverbucher / Ausleihautomat auch die Buchsicherung (sofern vorhanden) bedienen. Bei der in vielen Bibliotheken auslaufenden EM-Sicherung (elektromagnetisch über einen im Medium eingeklebten magnetisierbaren Metallstreifen) geschieht dies über die Ansteuerung eines Elektromagneten mit hörbarem Feedback an die Nutzenden („klack“). Bei RFID-Sicherung wird bei erfolgter Verbuchung ein Sicherungsbit auf dem RFID-Chip verändert. Aufgrund der größeren Geschwindigkeit dieses Vorganges geschieht dies ohne Feedback an die Nutzenden.

#### Rückgabeautomat / -sortierung

Ein separater Rückgabeautomat hat zum einen den Vorteil, dass die Prozesse Ausleihe und Rückgabe bei starker Nutzung entzerrt werden und zum anderen, dass eine Sortierung der zurückgegebenen Medien möglich ist. Die Medien werden von den Nutzenden auf ein Förderband gelegt und eingezogen (außer Reichweite des Nutzenden). In dieser Position wird der Barcode auf dem Medium oder der RFID-Chip gelesen. Wird keines der beiden erkannt, wird das Medium wieder zurückgegeben. Bei erfolgreicher Erkennung und Verbuchung im BMS (und anschließender Aktivierung der Buchsicherung) wird im BMS mithilfe der Signatur oder Mediennummer erfragt, wie das Medium sortiert werden soll. In den meisten BMS gibt es dazu Tabellen, die z.B. über die Anfänge von Signaturen oder anderen Kriterien (Bsp: „SN …“ in Wagen 3, „ist vorgemerkt“ in den Wagen x) arbeiten. Steht das Sortierziel fest, wird das Medium über Förderbänder zu dieser Stelle transportiert und abgeworfen. Das Ziel kann ein sog. Tray sein, ein oben offener Korb oder Wagen, oft mit einem gewichtgesteuerten Boden, damit die Medien nicht allzu tief fallen. Alternativ bieten immer mehr Hersteller sog. Ergocarts an, auf die die Medien so geschichtet werden, dass sie am Regal Rückenschonend aus einem Stapel entnommen und einsortiert werden können.

Üblicherweise gibt es am Rückgabeautomaten keine Authentifizierung.

Es gibt auch Rückgabeautomaten, die eine erneute Ausleihe des gerade zurückgegebenen Werkes an den gleichen Bibliotheksnutzenden ermöglichen. Dies ist in den Fällen sinnvoll, wenn die maximale Leihfrist / maximal mögliche Verlängerungen der Leihfrist erreicht ist und der Bibliotheksnutzende das Buch weiter nutzen möchte und das Medium nicht anderweitig bestellt ist.

#### Kassenautomat

Ein Kassenautomat ermöglicht die personalfreie Bezahlung der offenen Gebühren. Auch hier wird erst der Nutzungsausweis eingelesen und nach einer optionalen Passworteingabe die offenen Gebühren angezeigt. Die Gebühren können dann mit Bargeld oder Bargeldlos gezahlt werden. Auf eine Bargeldzahlung wird zunehmend verzichtet, da das Handling von Bargeld aufwändig und teuer ist.

#### Fernleihautomat

Aus einem Fernleihautomat können Fernleihen personalfrei an Nutzende ausgegeben werden. Da diese Bücher weder mit dem eigenen System der Bibliothek gesichert noch verbuchbar sind, muss eine separate Verbuchung durchgeführt werden. Die Nutzenden bekommen eine Nachricht, dass ihr bestelltes Medium in einem Fach mit der Nummer xy bereit liegt sowie eine PIN zur Öffnung dieses Faches. Sobald das Fach geöffnet wird, wird das Medium auf das Konto des Nutzenden verbucht. Auch eine Öffnung des Faches mit einem funkgesteuerten Nutzungsausweis statt der PIN ist möglich.

#### Sicherungsgates

Sicherungsgates erkennen unverbuchte Medien, die die Bibliothek verlassen. Die dafür übliche Technik war in den letzten Jahrzehnten die EM-Sicherung, also die Erkennung der Magnetisierung von metallischen Streifen, die in die Medien geklebt waren. Mit der Umstellung auf RFID geschieht die Buchsicherung über Funk, ein Sicherungsbit im Speicher der RFID-Chips wird untersucht. Bei EM-Sicherung ist der maximale Abstand zwischen zwei Gates zur halbwegs zuverlässigen Erkennung ca 90 cm und stellen somit eine Einschränkung des Zugangs, z.B. bei der Nutzung mit Rollstühlen, dar. Etwa der gleiche Abstand ist notwendig bei RFID-HF, bei RFID-UHF (Reichweite bis zu 10m) ist ein sehr großer Abstand möglich und somit der Verzicht auf eine Einengung des Ausgangs.

Bei Erkennung eines gesicherten (und nicht entliehenen) Mediums ertönt ein Warnton. Bei manchen Systemen wird das entsprechende Medium mit Titel und Cover auf einem Monitor angezeigt.

Sicherungsgates verhindern nicht Diebstahl, Diebe wählen andere Wege. Sicherungsgates verhindern das versehentliche Verlassen der Bibliothek mit unverbuchten Medien.

### Anbindung von Systemen über Schnittstellen

Ein BMS muss in der Lage sein, mit anderen Systemen automatisiert Daten auszutauschen. Diese Austauschprozesse betreffen folgende Szenarien

1. Bereitstellung von Konto- und Verfügbarkeitsinformationen, z.B. über [PAIA](https://verbundwiki.gbv.de/display/VZG/PAIA) und [DAIA](https://verbundwiki.gbv.de/display/VZG/DAIA)
2. Anbindung an Buchhaltungssysteme wie SAP oder HIS Haushalt-ERO
3. Anbindung an Tools für statistische Auswertungen (s.a. [Kapitel Statistik](#statistik))
4. Bereitstellung von bibliografischen Daten
5. Recherche in Fremddatenbeständen, z.B. über Z39.50
6. Schnittstellen zu Kataloganreicherungsdiensten (Buchcover)
7. Schnittstellen zu IDM-Systemen (s.a. [Kapitel IDM](#identity-management))
8. Schnittstellen zu einschlägigen Plattformen der jeweiligen Zielgruppen, zum Beispiel Lernmanagementsysteme

Die Systeme der neuen Generation verfügen in der Regel über Schnittstellen, über die sie in die bestehenden Informationsinfrastrukturen, d.h. die umgebenden Systeme, eingebunden werden können.

Eine Schnittstelle (engl. Interface oder manchmal auch API - application programming interface) bildet einen definierten Kommunikationsweg zwischen Verschiedenen Systemen als "Gesprächspartner". Im bibliothekarischen Universum gibt es für diese Fälle auch schon viele etablierte Austauschformate, etwa SIP2. Ein BMS "von der Stange" kann im Regelfall die üblichen Austauschformate unterstützen, sodass ein Austausch zwischen den gängigen Systemen einfach möglich ist. Hierzu zählen insbesondere der jeweilige Bibliotheksverbund, etwa zum Austausch von Metadaten oder für das verteilte Lizenzmanagement, aber auch nutzer\*innen-nahe Dienstleistungen, wie die Fernleihe.

### Nicht-bibliothekarische Schnittstellen

Ein BMS existiert im Regelfall nicht nur für sich oder nur im Kosmos der eigenen und anderer Bibliotheken, sondern ist auch in die lokalen IT-Strukturen eingebunden.

Ein gutes Beispiel ist der Einsatz eines BMS an einer Hochschule: Im Regelfall sind alle Mitglieder einer Hochschule auch gleichzeitig (potenzielle) Nutzer\*innen der Bibliothek. Die Daten der Mitglieder dieser Einrichtung werden an einer zentralen Stelle verwaltet und sollen durch andere Systeme, z.B. im Bibliothekssystem, durch Verknüpfung nachgenutzt werden. Dies ist die Rolle des Identity Managements (IDM).

#### Identity Management

Ein IDM (Identity Management System) ist ein System, mit dem die Basisdaten von Personen und Gruppen an zentraler Stelle verwaltet werden könne. Dies sind etwa persönlich Daten, Kontaktdaten und Organisationsstrukturen. Der Gedanke daher ist, dass alle relevanten Informationen nur an einer zentralen Stelle vorgehalten werden, und in anderen Systemen keine Dubletten erzeugt werden, die dann auch noch irgendwie synchronisiert werden müssten. Damit Personen in einem System eindeutig identifiziert werden können, existiert zumeist eine oder mehrere eindeutige IDs, etwa die Matrikelnummer eines Studierenden.

Das IDM hält im Regelfall mehr Daten über eine/n Nutzer/in bereit, als von den jeweiligen verbundenen Systemen benötigt werden. Beispielsweise könnte in einem IDM vorgehalten werden, dass eine Person Mitarbeiterin einer Hochschule ist, dass sie zu einer gewissen Fakultät der Hochschule gehört und dass sie zu einer bestimmten Arbeitsgruppe gehört. In der Kommunikation des BMS mit dem IDM ist jedoch nur die erste der Informationen relevant, etwa um die Ausleihkonditionen der Person festlegen zu können. Daher wird in der Kommunikation mit einem IDM im Regelfall auch auch gewisser Scope mitgegeben, damit nur die wirklich für das konsumierende System relevanten Informationen mitgegeben werden; Prinzip "Datensparsamkeit".

Ein IDM kann als Identity Provider zu einem Authentifizierungsdienst werden. Über diesen Dienst kann man dann unter Umständen ein Single Sign On realisieren, bei dem die Daten des Nutzenden nicht mehr an den Service oder Content Provider weitergegeben werden sondern nur noch eine Art Ticket, dass eine Erlaubnis regelt. Im Idealfall gilt diese einmalige Anmeldung dann für einige/viele Service-/Contentprovider, so dass der Nutzende sich nur einmal anmelden muss um viele Dienste zu nutzen.

Authentifizierungsprotokolle sind bspw.: [Shibboleth / SAML2](https://en.wikipedia.org/wiki/Shibboleth_(software)), [OpenID](https://de.wikipedia.org/wiki/OpenID)

Softwareprodukte für IDM sind: SAP (mit Plugins), Microsoft Active Directory, uvm

##### Speicherung von Nutzendenaccounts

Ein Account besteht aus den Kontaktdaten des Menschen sowie Authentifizierungsinformationen. Hier ist Datensparsamkeit nach DSGVO geboten. Für die Speicherung aller personenbezogenen Daten müssen die Notwendigkeiten oder rechtlichen Gründe nachgewiesen werden. Als Beispiel kann die Speicherung des Geburtsdatums angesehen werden. Wird für die Begründung für die Speicherung des Geburtsdatums die Prüfung der Volljährigkeit oder die Befähigung eines Seniorentarifes herangezogen, ist davon auzugehen, dass die Speicherung des Geburtsdatums nicht notwendig ist. Wird zur Begründung eine als notwendig erachtete Adressermittlung bei Behörden angegeben, ist die Speicherung der Geburtsdatum möglich, das eine Adressermittlung (zur Wiederbeschaffung vermisster Exemplare) möglich. Die Speicherung nutzungsbezogener Daten wie Verweise auf die ausgeliehenen Medien, angefallene Gebühren, offene Bestellungen und bestellte Digitalisate muss in der Regel nicht explizit begründet werden.

Sofern die übergreifende Institution über eine Datenbank zur Speicherung der Accounts verfügt ([IDM](#identity-management), Identity Management) ist eine Anbindung an diese sinnvoll. Dieses IDM enthält dann allerdings nicht notwendigerweise die externen Nutzenden.

Die technisch einfachste Lösung für Accounts der externen Bibliotheksnutzenden ist die Speicherung im IDM der übergeordneten Einrichtung, sofern vorhanden. Komplexer ist die Speicherung in einem separaten System, da dann bei Autorisierung u.U. mehrere Systeme abgefragt werden müssen.

Datenschutzbezogene Vorgehensweisen auch in Bezug auf personenbezogene und personenbeziehbare Daten von Nutzenden finden sich in Abschnitt [Datenschutz](#datenschutz) im Kapitel zum technischen Betrieb eines BMS.

#### Bezahlsysteme

Online-Payment, Kassensysteme/-automaten (siehe auch Kapitel [Kassenautomat](#kassenautomat))

#### E-Rechnung

E-Rechnungen müssen seit 2020 von Einrichtungen des Bundes, der Länder und Kommunen verarbeitet werden können. Der Umgang mit E-Rechnungen ist sehr unterschiedlich geregelt. Zum Teil nehmen Einrichtungen nur noch an einer zentralen Stelle E-Rechnungen entgegen. In anderen Einrichtungen werden E-Rechnungen dort entgegen genommen, wo die Bestellungen ausgelöst wurden. Es gibt verschiedene Formate in der eine E-Rechnung übermittelt werden kann (PDF, XML oder direkt per [EDIFACT](https://de.wikipedia.org/wiki/EDIFACT)).

Elektronische Rechnungen kommen immer dann ins Spiel, wenn Bestellvorgänge von neuen Medien über das BMS abgewickelt werden. In diesem Zusammenhang entstehen Rechnungen von Lieferanten, die von der Bibliothek oder ihrer Organisation zu begleichen sind.

Ohne eine „E-Rechnungs-Workflow“ würde dies bedeuten, dass Rechnungen der Lieferanten bei der Bibliothek eingehen, einem Bestellvorgang zugeordnet werden müssen, von der jeweiligen Rechnungsstelle beglichen und schließlich wieder im BMS „abgehakt“ werden müssen. Diese repetitiven Workflows lassen sich mittlerweile weitgehend automatisieren. Das BMS ist in der Lage elektronisch übermittelte Rechnungsdaten automatisiert den jeweiligen Bestellprozessen zuzuordnen Bei einer gleichzeitigen Anbindung eines elektronischen Rechnungswesens z.B. über SAP können auch die Zahlungsinformationen automatisiert zugeordnet werden und somit ein Bestellvorgang komplett automatisiert abgeschlossen werden.

#### Statistik

Mit dem Begriff „Statistik“ können verschiedene Dinge im Rahmen eines BMS gemeint sein, etwa Betriebsstatistiken, wie die Rechnerauslastung eines Servers, auf dem das BMS betrieben wird. In diesem konkreten Fall ist jedoch mit „Statistik“ gemeint, dass die Nutzungsdaten des BMS in eine Form gebracht werden können, in der Mitarbeiter\*innen der Bibliothek Informationen ziehen können, die zur Dokumentation, zum Reporting oder zur weiteren Arbeit benutzt werden können.

Beispiele für Statistiken sind ganz klassische die Ausleihzahlen einer Bibliothek, ggf. aufgeteilt nach verschiedenen Themen oder Fächern, die den Bedarf der an bestimmten Medien preisgeben. Dies könnte für die Leitung einer Bibliothek relevant sein, oder für das Budgetmanagement der Einrichtung. Ein anderes Beispiel wären detaillierte Ausleih- und Benutzungsstatistiken, sowie konkrete Informationen zum Bestand. Mit einer solchen Datenlage können Fachbibliothekar\*innen etwa gezielt Bestandsmanagement betreiben.

Manche LMS haben Statistikfunktionen schon mit dabei. Andere halten ihre Daten in einer Datenbank und diese müssen aktiv exportiert werden. Wieder andere bieten entsprechende Schnittstellen, über die statistische Daten exportiert werden können.

Je nach Anforderung an den Umfang und an die Arbeit, die mit statistischen Auswertungen erfolgen soll, kann die Entscheidung fallen, die aus dem BMS kommenden Daten einfach nur in eine Tabellenkalkulation zu exportieren, oder eine speziell auf die statistische Datenanalyse zugeschnittene Statistik-Software einzuspielen. Ergo kommen hier Softwares wie Excel, BibControl oder gar komplexe Statistik-Plattformen wie SPSS in Frage. BMS wie ExLibris Alma bringen hierbei schon eigene Statistik-Module mit, die eine externe Lösung überflüssig machen.

Alma (und andere BMS) können automatisiert oder manuell [COUNTER](https://www.projectcounter.org/counter-sushi/)-Reports für statistische Daten der Nutzung digitaler Medien importieren.

## Technischer Betrieb

|  |
| --- |
| Hinweis |
| Der technische Betrieb eines BMS umfasst die Installation, die [lokal](#lokale-installation) oder als [gehostete Variante](#hosting) bzw. in der [Cloud](#cloud) erfolgen kann. [Kosten](#kosten) entstehen dabei für Lizenz- und Wartungsverträge sowie für Betriebsressourcen. Für den Betrieb sind weiter das [Monitoring](#monitoring) sowie die Aspekte der [IT-Sicherheit](#it-sicherheit), [Backup](#backup-und-rollback) und [Datenschutz](#datenschutz) zu berücksichtigen. |

### Betriebsmodelle für serverbasierte Software

Ein BMS kann auf verschiedene Arten betrieben werden. Die Betriebsarten unterscheiden sich bezüglich Installation, Kosten, Pflege und Wartung sowie Backup und Support.

#### Lokale Installation

Bis etwa 2010 war diese Betriebsart der Normalfall: Eine Einrichtung erwarb die Lizenz für eine (Server-)Software, entweder als Einzelkauf oder im Abo, und installierte diese auf eigenen Servern, z.B. im Serverraum der Bibliothek. Im Fachjargon spricht man auch von einer „on-premise“ Installation.

In diesem Modell kümmert sich die Einrichtung selbst um Installation und Updates. Folglich erfordert dieses Modell höheren Personaleinsatz und kann dazu führen, dass bei einem personellen Engpass eine Software länger betrieben bzw. nicht aktualisiert wird, als eigentlich ratsam wäre. Auch muss sich die Einrichtung um grundlegende Dinge, wie Backups und Ausfallsicherheit selbst Gedanken machen.

Auf der anderen Seite bietet dieses Modell der Einrichtung die meiste Kontrolle über die eingesetzte Software - etwa hinsichtlich nötiger Erweiterung oder Anpassung - und macht sie damit weitgehend unabhängig von äußeren Einflüssen.

#### Hosting

In diesem Betriebsmodell wird die Ebene der Rechenkapazität bzw. Serverhardware an einen Dienstleister ausgelagert. Der Dienstleister kann hierbei etwa das Rechenzentrum einer Universität oder des angeschlossenen Bibliotheksverbundes sein, oder ganz allgemein jeder kommerzielle Betreiber eines Rechenzentrums, bei dem Kapazitäten erworben werden.

Sämtliche Betriebsfragen, wie Backups und Ausfallsicherheit der eingesetzten Hardware können an diesen Anbieter delegiert werden. Im Falle des Hostings durch einen Bibliotheksverbund entfallen möglicherweise auch Einrichtung, Installation und Upgrades des BMS. Die Betriebskosten müssen beim Verbund kalkuliert werden, was jedoch durch das Hosting für mehrere Einrichtung besser skaliert.

#### Cloud

Bei diesem Betriebsmodell, das manchmal auch als SaaS (Software as a Service) bezeichnet wird, liegt der fachlich und technische Betrieb beim Anbieter bzw. Dienstleister des BMS und die Einrichtung nutzt lediglich eine Installation/Instanz des BMS. Dies ist insbesondere bei webbasierten BMS die bevorzugte Betriebsart, stellt aber erhöhte Anforderungen an die Anbindung lokaler Endgeräte (s.a. z.B. auch [Automatisierung](#selbstverbucher-ausleihautomaten)), weil dabei eine sichere und stabile Verbindung zwischen den lokalen Automatisierungsgeräten und dem entfernt gehosteten System hergestellt werden muss.. Die Einrichtung ist weder für die Wartung der eingesetzten Hardware noch für die Pflege der genutzten Software zuständig.

In der Praxis kann sich ein solches Betriebsmodell als komfortabel erweisen, da keine Personalressource für allgemeine Tätigkeiten des IT-Betriebs oder spezielle Bibliotheks-IT-Tätigkeiten benötigt werden. Gerade für kleine Einrichtungen kann dies ein guter Weg sein, möglichst personalsparend ein BMS einzusetzen. Eine Kostenersparnis ist bei einer Vollkostenrechnung aber nicht unbedingt zu erwarten. Je nach Größe der Einrichtung oder basierend auf der Anzahl der Endnutzer\*innen führt ein solches Betriebsmodell meist zu Abonnementkosten.

### Kosten

Die Anschaffungskosten eines BMS machen nur einen kleinen Teil aus. Wichtiger ist, sich über folgende Kosten klar zu werden:

* Personalkosten für den laufenden Betrieb
* Lizenzkosten und Wartungsverträge der Software
* Betriebsressourcen, wie z.B. Serverraum, Energieverbrauch, Wartung, Backuplösungen

Personalkosten und Ressourcen richten sich hauptsächlich nach Art der Installation (Lokal, Hosting oder Cloud). Lizenzkosten sind teilweise nach Größe der Einrichtung gestaffelt, d.h. sie richten sich nach Anzahl der verwalteten Medien und/oder Endnutzer\*innen.

Insbesondere der Punkt Personalkosten kann zu einem Engpass bzw. Risiko werden, denn in vielen Fällen zeigt sich, dass einige wenige Personen durch ein BMS gebunden werden und gleichzeitig auch die einzigen sind, die das System in der Tiefe bedienen können. Wirklich kritisch wird es, wenn nur eine einzige Person diese Rolle erfüllt. Je mehr Verantwortung beim Betrieb auf das Personal vor Ort fällt (lokaler Betrieb), desto wichtiger wird dieser Aspekt. Selbst bei der Nutzung eines Cloud-BMS ist davon auszugehen, dass für die fachliche Administration der Software Personal dauerhaft gebunden ist. Bei dieser Betriebsmethode gibt der Anbieter meistens den Updatezeitpunkt vor, insofern müssen unter Umständen Workflows in der Bibliothek aufgrund von Änderungen in der Software durchgeführt werden, ohne dass man die zeitliche Planung dafür in der Hand hat.

Um Personalengpässe zu vermeiden, ist es sinnvoll, Einführungsprozesse nur in einer Expertengruppe durchzuführen und Verantwortlichkeiten auf mehrere Schultern zu verteilen (Ausfallsicherheit, Urlaubsvertretung usw). Auch die gute Dokumentation teils komplexer Zusammenhänge sollte bedacht werden, damit Fachwissen nicht nur in den Köpfen einiger weniger Mitarbeiter\*innen schlummert.

### Installation & Updates

Zur Einrichtung eines BMS gehört:

* **Installation auf einem Server:** erfordert i.d.R. Kenntnisse in Systemtechnik (Hardware, Server, Kommandozeile...). Wenn Hosting durch Drittanbieter geleistet wird (Cloud, Dienstleister wie Verbundzentrale o.A.), verändert sich diese Aufgabe. Sie entfällt, wenn der Hoster spezialisiert auf das Hosting von BMS ist (bspw. Verbundzentrale), sie wird geringer, wenn der Hoster eher allgemein aufgestellt ist.
* **Konfiguration/Parametrisierung:** Teilweise über Administrator-Oberfläche möglich, teilweise nur über Konfigurationsdateien. Erfordert vor allem Kenntnisse der eigenen IT-Infrastruktur und der verwendeten Schnittstellen und Formate. Die Grenzen zwischen Konfiguration und Programmierung eigener Erweiterungen sind fließend. Zu beachten ist auch die Migration bestehender Daten in das neue System.

Nach Einrichtung werden BMS laufend erweitert. Fehler werden behoben und neue Funktionen kommen hinzu. Die Aktualisierung kann je nach Produkt agil in kleinen, häufigen Schritten erfolgen (monatlich, wöchentlich oder häufigere Updates) oder in längeren Zeitabschnitten (oft quartals- oder halbjahres-weise).

### Open Source

Wird ein System auf Open-Source-Basis eingesetzt, sollte eine Verständigung darüber erfolgen, ob und unter welchen Bedingungen lokale Anpassungen am System auch der Community zur Verfügung gestellt werden. Hierzu müssen die Lizenzbedingungen des Systems geprüft werden.

### Laufender Betrieb

Während des laufenden Betriebs ist es wichtig, sich über den aktuellen Betriebszustand des Systems ein klares Bild machen zu können. Dieser „Statusbericht“ kann sich über alle Ebenen des Systems ziehen: Wie viel Speicherplatz ist noch frei? Ist das System für alle Nutzenden erreichbar? Sind verbundene Systeme verfügbar und betriebsbereit? Je nach Betriebsmodell werden diese Fragestellungen durch klassisches IT-Monitoring abgedeckt, benötigen teilweise aber auch bibliotheksspezifische Lösungen.

#### Monitoring

Monitoring-Lösungen für den Betrieb von IT-Infrastrukturen sind beispielsweise [Check\_MK](https://checkmk.com/) oder [Prometheus](https://prometheus.io/). Diese Anwendungen bieten eine kontinuierliche Überwachung von Systemen anhand definierter Metriken und warnen die Administratoren aktiv, wenn definierte Werte bestimmte Grenzen überschreiben.

Die Nutzung einer Monitoring-Lösung wird umso relevanter, je mehr Betriebsverantwortung für das BMS bei der Einrichtung liegt. Beim Cloud-BMS liegen zwar viele der Verantwortungen beim Betreiber der Software, trotzdem sollte zumindest die reine Verfügbarkeit des Systems auch von der nutzenden Einrichtung überwacht werden.

#### Notfallbetrieb

Bzgl. der Themen Support, Wartung & IT-Sicherheit, als auch Fehlersuche und -vorbeugung, unterscheiden sich die Aufwände für die Einrichtung je nach gewähltem Betriebsmodell erheblich. Jedoch weichen die zu nutzenden Prinzipien bei diesen Themen nicht grundlegend ab zu anderen zu wartenden Systemen in der IT-Welt.

Dazu gehören Maßnahmen zur Aufrechterhaltung des Bibliotheksbetriebs im Notfallbetrieb. Dies kann ein temporärer Offlinebetrieb des Systems sein. In diesem Fall werden die Prozesse mit den Daten abgewickelt, die zum Zeitpunkt des Offline-Gangs im System vorhanden waren. Wenn das System wieder online geht, muss gewährleistet werden, dass Änderungen an den Daten aus der Offlinezeit nachvollzogen werden (Beispiele: Ausleihen, Erwerbungen, Rechnungsbearbeitung, Nutzerdatenänderungen). Im Idealfall erledigt das die genutzte Komponente oder das BMS selbst.

Bei lokalen Installationen sollte man je nach Größe der Einrichtung ebenfalls über ein Spiegelsystem des BMS nachdenken. Dieses wird parallel auf dem aktuellen Stand gehalten und kann einspringen, wenn das laufende BMS ausfällt.

Es empfiehlt sich in jedem Fall neben dem Einsatz eines Produktivsystems mindestens eine Test-Instanz und ggf. eine oder mehrere Entwicklungs-Instanzen des BMS zu betreiben. So können neue Funktionen schneller umgesetzt werden ohne den laufenden Betrieb durch unerwartete Fehler zu gefährden.

Zu beachten ist weiterhin, die Nutzenden des BMS (intern als auch extern) bei Problemen zu informieren. Dabei sind vor allem von Bedeutung, welche Interaktionen nicht mehr möglich sind, ob es alternative Möglichkeiten für die Nutzenden gibt und wann das System voraussichtlich wieder zur Verfügung steht.

### IT-Sicherheit

Um ein BMS vor den zunehmenden Angriffen durch böswillige Akteure (Hacking, Malware, Ransomware) abzusichern, können die folgenden Empfehlungen als Grundlage dienen (Breeding, Marshall 2022):

* Die Infrastruktur um das BMS herum sollte durch starke Sicherheitsvorkehrungen getragen werden.
* Die Gefahr kurzfristig entstehender Sicherheitslücken sollte nicht unterschätzt werden.
* Cloud-basierte Systeme sollten aktiv überwacht und der Überblick behalten werden.
* Anbieter sollten aufgefordert werden, die Konzepte ihrer Sicherheitsvorkehrungen offenzulegen.
* Gerade Administrator\*innen sollten ihre Zugänge gesondert absichern.
* Es sollte sichergestellt werden, dass jede Software stets auf dem aktuellen Stand ist, sowohl auf den Arbeitsplatz-PCs als auch den Servern.

Allgemein gilt auch immer der Grundsatz: „Bleiben Sie wachsam, in Bezug auf ungewöhnliche Ereignisse auf Ihren IT-Systemen“.

### Backup und Rollback

Für den Fall, dass der Betrieb eines BMS lokal erfolgt, ist es wichtig, dass sich die Einrichtung über Backup und Rollback der Software Gedanken macht. Da dies ein generelles Thema des Betriebs von IT-Systemen ist, wird im Folgenden auf die Spezifika für BMS eingegangen, und Themen wie das Backup von Servern lediglich angerissen.

Folgende Aspekte sollten im Rahmen von BMS besondere Beachtung finden:

* Definition von Backup-Zyklen: wie oft werden welche Daten in welchem Umfang auf welche Art gesichert? Es können hier durchaus verschiedene "Sicherungsaspekte" mit unterschiedlichen Zyklen definiert werden.
* Definition des Umfangs der Sicherung. Sollen die Daten komplett gesichert werden, sollen nur Veränderungen gesichert werden? Wichtig für diese Entscheidung ist die Frage, wie schnell ein System wiederhergestellt werden kann/soll.
* Ein off-site-Backup sollte in die Überlegungen einbezogen werden, also ein kompletter Satz einer Sicherung, der außerhalb der Institution gelagert wird. Dabei ist der Datenschutz zu berücksichtigen, u.U. müssen die Sicherungsdaten daher verschlüsselt werden.
* Sicherung von Daten, die bei rechtlichen Fragen von Relevanz sein können: Bsp. Rechnungen, Ausleihen, Mahnungen.

Ganz allgemein ist die Frage zu klären, wer Verantwortung für die Einrichtung, Durchführung und die regelmäßige Kontrolle der Sicherungen hat. Der letzte Punkt meint hierbei einerseits das Monitoring der erfolgreichen regelmäßigen Ausführung von Sicherungen, aber auch der Test der erstellten Sicherungen, etwa durch periodisches Einspielen auf einer Testinstanz des BMS.

### Zusammenspiel Hard- und Software

Das BMS steht in der IT-Landschaft einer Bibliothek im Regelfall nicht alleine, sondern kommuniziert mit anderen Hard- und Softwaresystemen. Hierzu gehören beispielhaft (s.a. Kapitel [Automatisierung)](#automatisierung-und-selbstbedienung):

* Lesegeräte: (Barcode-)Scanner und Chip-Lesegeräte für Benutzungsausweise und/oder Medien
* Selbstverbucher für die Ausleihe und/oder Rückgabe
* Rückgabeautomaten, die ggf. auch eine automatische Vorsortierung von Medien übernehmen
* Sicherungsgates zur Detektion nicht entliehener Medien an Ein- und Ausgängen
* Drucker zur Erstellung von Quittungen, Ausweisen, Labeln, usw.

Für den Fall, dass die externen Systeme nicht lokal an einem Computer angeschlossen sind, sondern über das Netzwerk der Einrichtung angebunden sind, gibt es vielfach etablierte bibliothekarische Schnittstellen (APIs), etwa [SIP](https://en.wikipedia.org/wiki/Standard_Interchange_Protocol), oder man setzt auf moderne, allgemeine API-Standards wie [REST](https://de.wikipedia.org/wiki/Representational_State_Transfer).

### Datenschutz, User-Tracking, Analytics

Innerhalb der EU gilt seit 2018 die Datenschutz-Grundverordnung (DSGVO), nach der personenbezogene Daten grundsätzlich zu schützen sind.

Im Kontext eines BMS sind die anfallenden personenbezogenen Daten etwa:

* durch Nutzer\*innen bei der Anmeldung angegebenen Daten für den Bibliothekszugang
* mit dem Nutzer\*innen-Konto verbundene Ausleihvorgänge und Mahnhistorien
* die Protokolle (Logs) über Online-Zugriffe auf das BMS (z.B. IP-Adresse, Seitenaufrufe)

Um den Schutz personenbezogener Daten gewährleisten zu können, gibt es verschiedene Ansätze:

* Verschlüsselung: Daten werden auf verschlüsselten Servern gespeichert, ebenso ist die Übertragung Ende-zu-Ende verschlüsselt
* Separierung: Personendaten werden getrennt von nicht-sensiblen Daten gehalten (siehe [IDM](#identity-management))
* Pseudonymisierung: Nutzer\*innen-Daten werden mit Pseudonymen präpariert, sodass sie nicht mehr oder nur unter großem Aufwand den einzelnen Personen zuzuordnen sind
* Anonymisierung: Daten werden derart verändert, dass sie nicht rückverfolgbar sind (z.B. Maskierung IP-Adressen)

Die Entscheidung zur Verschlüsselung und Separierung von Daten sollte bereits im Vorfeld des Betriebs eines BMS getroffen werden.

Die Pseudonymisierung und Anonymisierung kann auch im Laufe der Erhebung der personenbezogenen Daten zur Anwendung kommen, sofern bestimmte Daten nicht mehr für einen konkreten Zweck erforderlich sind.

Leider sind personenbezogene Daten für Bibliotheks-Statistiken oft notwendig (siehe [Kosten](#kosten)). In diesem Fall sollten ebenfalls pseudonymisierte oder anonymisierte Datensätze zur Grundlage genommen werden.

Wenn ein BMS durch einen externen Anbieter gehostet wird (siehe [Betriebsmodelle für serverbasierte Software](#X3a5ca0018f511a6e73c5bcce2da503f97a39d48)), muss Folgendes sichergestellt sein:

* Die Verschlüsselung der Datenübertragung (Ende-zu-Ende-Verschlüsselung)
* Betrieb und Steuerung der Server innerhalb der EU ([DSGVO](https://de.wikipedia.org/wiki/Datenschutz-Grundverordnung))
* Der Ausschluss von User-Tracking durch Ad-Tech (Werbe-Netzwerke)
* Der Abschluss eines [Datenverarbeitungsvertrags im Auftrag](https://de.wikipedia.org/wiki/Datenverarbeitung_im_Auftrag)

In der Kombination eines IDM mit einem cloudbasierten BMS außerhalb der EU wäre denkbar, die personenbezogene Daten dort in pseudonymisierter Form speichern zu lassen oder Personendaten von nicht sensiblen Daten zu trennen.

Für alle personenbezogenen und personenbeziehbaren Daten sind Lösch- oder Anonymisierungsfristen festzulegen. Die Anonymisierungsfristen ergeben sich aus den Vorgaben der DSGVO und müssen betrieblichen und rechtlichen Aspekten genügen. So ergeben sich Fristen für die Speicherung von Daten über Gebühren (Entstehung, Bezahlung, ...) aus den Landeshaushaltsordnungen oder anderen für die Einrichtung maßgeblichen Regelungen. Betriebliche Gründe für die Länge von Speicherfristen von personenbezogenen und personenbeziehbaren Daten können sich aus Fristen für Einsprüche ergeben.

Die über die vergangenen Jahrzehnte geschehenen sukzessiven Aufkäufe kleinerer BMS-Service-Provider durch einige wenige große kommerziellen Bibliotheksdienstleister hat ganze Firmenkonglomerate entstehen lassen, die inzwischen den Bibliotheksmarkt dominieren. Einige von ihnen, die Dienste für wissenschaftliche Bibliotheken anbieten, wandeln sich in den letzten Jahren zu Data-Analytics-Konzernen. In diesem Zuge präparieren sie ihre cloud-basierten BMS-Lösungen mit Trackern, die Verhaltensprofile über die Nutzenden erstellen. Durch die ebenfalls seitens der Anbieter gestellten Zugangsauthentifizierungssysteme wird versucht, zusätzlich eine möglichst hohe Personalisierung bei der Erstellung einzelner Profile zu erreichen. Die dabei entstehenden Datenflüsse werden für gewöhnlich nicht transparent gemacht (Siems 2022). Der Einsatz solcher Analytics-Technologien unterminiert die Integrität konventioneller IDM-Systeme und tangiert somit nicht nur datenschutzrechtliche Belange, sondern auch die IT-Sicherheit. Idealerweise sollte bereits vor der Anschaffung einer BMS-Lösung abgeklärt werden, ob solche Analytics-Technologien eingesetzt werden. Im Zweifelsfall sollte immer der\*die lokale Datenschutzbeauftragte oder IT-Sicherheitsbeauftragte hinzugezogen werden.

## Zusammenfassung und Ausblick

Ein BMS ist im Normalfall kein statisches System - vielmehr muss es aufgrund der sich verändernden Bedürfnisse einer Bibliothek und deren Nutzenden stetig angepasst werden.

Der Import, Export oder auch die Zusammenführung von Daten erfordert klar definierte Metadaten und Schnittstellen für den freien Austausch aus gut nachnutzbaren Quellsystemen. Dies ist vor allem erforderlich bei der aktuell stärkeren Entwicklung hin zu Open Data und öffentlicher Datennutzung. Die Integration und Interaktion mit anderen Informationssystemen nimmt also zu. Vor allem herkömmliche BMS der zweiten Generation kommen hier schnell an ihre Grenzen.

# Discovery & Retrieval

|  |
| --- |
| Hinweis |
| Diese Hauptkapitel wird erst noch erstellt. |

# Daten und Metadaten

Zur Sammlung und Bereitstellung von Informationen werden von Bibliotheken Dokumente unterschiedlichster Form (Bücher, Filme, Forschungsdaten, Objekte…) nachgewiesen. Zur Verwaltung dieser Medien werden die Dokumente mit **Metadaten** beschrieben. Neben diesen Metadaten beinhalten bibliothekarische Informationssysteme zunehmend auch Dokumente selbst als **digitale Inhalte** wie sogenannte Volltexte, Digitalisate und Forschungsdaten (*siehe noch offene Kapitel Forschungsnahe Dienste und Digitalisierung*). Viele der im Folgenden beschriebenen Grundlagen zu Eigenschaften, Arten und Verarbeitung von Daten gelten sowohl für Metadaten als auch für digitale Inhalte.

## Daten

Im Wesentlichen bestehen Daten im Sinne dieses Buchs aus einer Folge von Bits. Abgesehen von ihrer Anzahl in Bytes lässt sich auf dieser Ebene allerdings nichts weiter über Daten sagen. Uns interessiert daher mehr, für was die Daten stehen – beispielsweise für eine Jahreszahl, ein Bild oder für den Titel eines Dokumentes. Dabei besteht ein Unterschied ist der zwischen

* der Struktur von Daten (**Syntax**)
* und der Bedeutung von Daten (**Semantik**).

Zur Interpretation von Daten dienen **Kodierungen** in Form von Datenformaten und Identifikatoren. Wo genau jeweils die Grenze zwischen Syntax und Semantik liegt, hängt davon ab, auf welcher Ebene und mit welcher Kodierung Daten betrachtet und verarbeitet werden (siehe [Tabelle 6.1](#tbl-daten-ebenen)):

Tabelle 6.1: Beispiele für Syntax und Semantik von Daten auf verschiedenen Ebenen

| Datenebene | Bedeutung | Kodierung |
| --- | --- | --- |
| 1100001 | Die Zahl 97(64+32+1) | Byte als Zahl |
| 97 | Der Buchstabe a | ASCII oder Unicode |
| a | Unterfeld für Haupttitel | Feld 021A im PICA+ K10plus-Format |
| a | Unterfeld für Umfangsangabe | Feld 300 im MARC21 Format |

Ein Großteil der [Datenverarbeitung](#datenverarbeitung) besteht darin darin Daten, zum Beispiel im Rahmen von [ETL-Prozessen](#etl-prozess), von einer Kodierung in eine andere Kodierung zu überführen, um sie anschließend leichter interpretieren zu können. Bei der Konvertierung von Daten von einem in ein anderes Datenformat reduziert sich das implizite Datenmodell der Konvertierung auf den kleinsten gemeinsamen Nenner beider Formate.

Zur Beschreibung von Daten dienen

* formale **Schemas** auf Ebene der Syntax
* und **Datenmodelle** auf Ebene der Semantik.

Leider liegen beide oft nicht explizit vor, sondern müssen anhand von Beispielen, Anwendungen und Dokumentation mühsam ermittelt werden. Im Idealfall entsprechen Daten einem klar definierten Datenformat.

## Datenformate

Datenformate definieren eine Struktur, die sich in einer oder in mehreren austauschbaren Syntax-Varianten ausdrücken lässt und deren Bedeutung durch ein Datenmodell festgelegt ist. Beispielsweise definiert der Unicode-Standard eine Menge von Schriftzeichen (Buchstaben, Sonderzeichen, Emojis…) und verschiedene Verfahren, um Zeichenketten in Bytes zu kodieren (UTF-8, UTF-16…). Syntax-Varianten werden auch als **Serialisierung** bezeichnet. Die meisten Datenformate haben nur eine Serialisierung, so dass Format und Syntax meist synonym verwendet werden. Einzelne Syntax-Elemente entsprechen Bestandteilen im Datenmodell (siehe [Tabelle 6.2](#tbl-xml)), daher werden in der Beschreibung von Daten auch diese beiden Ebenen meist nicht sauber getrennt.

Tabelle 6.2: Einige Bestandteile des XML-Format

| XML-Syntax | XML-Modell |
| --- | --- |
| <name /> oder <name>...</name> | XML-Element |
| name="Inhalt" | XML-Attribut |
| <!-- ... --> | Kommentar |

Als Faustregel kann gelten, dass bei statischer Betrachtung von Daten der Bezug auf ihre Syntax sinnvoll ist, während zur Verarbeitung von Daten eher auf die Bestandteile ihrer Modelle Bezug genommen werden sollte.

Datenformate lassen sich grob in zwei Kategorien unterteilen:

* **Strukturierungssprachen** wie CSV, XML, JSON und RDF ermöglichen es, Daten in kleinere Einheiten zu unterteilen und miteinander in Beziehung zu setzen. Die Sprachen basieren auf allgemeinen Ordnungsprinzipien (Felder, Tabellen, Hierarchien, Netzwerke…) und ihre Modelle haben darüber hinaus keine eigene Semantik. Die einfachste Strukturierungssprache ist das Prinzip der Zeichenkette.
* **Anwendungsformate** legen die Struktur von Daten für konkrete Arten von Inhalten fest ([Metadatenformate](#metadatenformate) zur Beschreibung von Dokumenten, Bildformate für Bilder…). Ihre Modelle verweisen letztendlich auf reale Objekte und Eigenschaften. Viele Anwendungsformate sind ihrerseits mittels einer Strukturierungssprache kodiert, zum Beispiel basiert das DataCite-Format zur Beschreibung von Forschungsdaten auf dem XML-Modell.

Darüber hinaus gibt es besondere Formate, deren Anwendung in der Verarbeitung von Daten liegt. Neben Programmiersprachen, die selbst nicht als Datenformate betrachtet werden, sind dies folgende Sprachen:

* **Schemasprachen** wie XML Schema, JSON Schema und nicht zuletzt *reguläre Ausdrücke* dienen der formalen Beschreibung der Syntax von Datenformaten. Dabei bezieht sich jede Schemasprache auf eine Strukturierungssprache (*XML Schema* für XML-Formate, *Avram* für feldbasierte Formate…).
* **Abfragesprachen** dienen dem Verweis auf einzelne Teile von Datensätzen. Sie beziehen sich ebenfalls immer auf eine Strukturierungssprache (zum Beispiel XPath für XML, JSON Path für JSON…) und sind für die Verarbeitung von Daten notwendig.
* **Modellierungssprachen** helfen zur Beschreibung von Datenmodellen. Die häufigsten Modellierungssprachen basieren auf dem Entity-Relationship-Modell (*ERM*). Da zwischen Syntax und Semantik irgendwann die reine Datenebene verlassen werden muss, sind die wichtigsten Mittel zur Datenmodellierung allerdings Diagramme und Beschreibungen in natürlicher Sprache.

Die Verwendung von Schema-, Abfrage- und Modellierungssprachen hilft, viele häufige Fehler bei der (Meta)datenverarbeitung zu vermeiden. Ein Beispiel hierfür ist das *Resource Description Framework* (*RDF*) mit dazu gehörigen Schemasprachen (*SHACL*/*ShEx*), Abfragesprachen (*SPARQL*) und Modellierungssprachen (*RDFS*/*OWL*). In anderen Fällen wird aus Mangel an Werkzeugen und Kenntnissen auf spezielle Datensprachen verzichtet und stattdessen auf allgemeine Programmiersprachen zurückgegriffen.

Eine ausführlichere Beschreibung von Datenformaten mit bibliothekarischem Schwerpunkt bietet die Datenbank [format.gbv.de](https://format.gbv.de/).

|  |
| --- |
| Tipp |
| **Reguläre Ausdrücke** sind das gängigste Mittel zur Beschreibung der Syntax von Daten. Gleichzeitig können mit ihnen Zeichenketten nach Mustern durchsucht werden. Ein regulärer Ausdruck für die Syntax einer ISBN-13 mit optionalen Trennstrichen ist beispielsweise:  (97[89])-?([0-9]{1,5})-?([0-9]+)-?([0-9]+)-?[0-9]  Üblicherweise decken Schemasprachen nicht alle Aspekte eines Datenformat ab: so lässt sich die Korrektheit der abschließenden Prüfziffer ([0-9]) nicht mit einem regulären Ausdruck überprüfen. |

In der Praxis werden Daten in einem Datenformat zusätzlich durch anwendungsspezifische Auslegungen und Einschränkungen geprägt, darunter Format-Varianten, Anwendungsprofile, Erfassungsregeln und -praxis.

## Datenerfassung

Traditionell werden bibliothekarische Metadaten durch Katalogisierung erstellt. Die Verwaltung der Katalogdaten erfolgt entweder lokal oder gemeinsam in einer **Verbunddatenbank**. Der Vorteil der Verbundkatalogisierung liegt darin, dass jedes Dokument nur einmal zentral beschrieben werden muss, während bei lokaler Katalogisierung durch *Fremddatenübernahme* nur zum Teil auf vorhandene Kataloge zurückgegriffen werden kann.

Im Idealfall sollte die Erfassung nach *Autopsie*, also auf Grundlage des vorliegenden Werkes, durch geschultes Personal und nach etablierten Regelwerken (*Katalogisierungsrichtlinien*) erfolgen. Um möglichst viele Publikationen zu erfassen, wird jedoch zunehmend auch auf anderweitig erfasste Metadaten von Verlagen, Repositories und aus anderen Quellen zurückgegriffen. Dazu müssen Daten unterschiedlicher Erschließungstiefe und -qualität im Rahmen von [ETL-Prozessen](#etl-prozess) gesammelt, analysiert und mit vorhandenen Daten vereinheitlicht werden. In jedem Fall muss beachtet werden, dass sich Regeln und Umstände, nach denen Daten erfasst werden, mit der Zeit ändern können (beispielsweise der Umstieg der Erfassungsregeln von *RAK* auf *RDA*) und dass das Ergebnis auch davon abhängt, wie gut überprüft werden kann, inwieweit die Anforderungen an die Daten sind.

Darüber hinaus gibt es Verfahren zur automatischen Erstellung von Metadaten aus vorhandenen Dokumenten, beispielsweise zur Erkennung und Auswertung von Literaturangaben und zur thematischen Einordnung von Dokumenten. Mit diesen Verfahren lassen sich zwar größere Mengen von Daten erfassen, es muss aber immer mit einer gewissen Fehlerrate gerechnet werden.

Welche Art und welcher Umfang von Fehlern und Uneinheitlichkeiten bei der Datenerfassung tolerierbar sind, hängt letztlich davon ab, wozu die Daten erfasst werden: so gelten beispielsweise für eine historische Bibliographie andere Maßstäbe als für einen Suchindex.

Nicht zuletzt sollte bedacht werden, dass Geschwindigkeit und Qualität von Datenerfassung auch von der Usability der Werkzeuge abhängen, mit denen Daten erstellt, bearbeitet und *analysiert* werden können.

## Identifikatoren und Normdaten

Ein wesentlicher Teil von Daten besteht aus Identifikatoren (IDs wie Nummern, Codes…) zum Verweis auf externe Objekte oder an anderer Stelle verwaltete Informationen. Identifikatoren ermöglichen die eindeutige Referenzierung gleicher Dinge in unterschiedlichen Kontexten, so dass Daten aus verschiedenen Quellen miteinander abgeglichen und kombiniert werden können.

### Identifikator-Systeme

Neben eher intern genutzten Datensatz-Identifikatoren (z.B. die *PPN* des Bibliothekssystems PICA oder die ZDB-ID der Zeitschriftendatenbank) sind vor allem international standardisierte Identifikatoren von Bedeutung. Beispiele für solche Identifier-Systeme mit Relevanz für bibliothekarischer Daten sind die nachfolgenden:

* Die *International Standard Book Number* (**ISBN**) wird von Verlagen für Bücher vergeben. Seit 2007 ist die 13-stellige ISBN Teil des EAN-Barcode-Systems.
* Die *International Standard Serial Number* (**ISSN**) identifiziert Zeitschriften und Schriftenreihen.
* Der *Digital Object Identifier* (**DOI**) identifiziert digitale Publikationen in elektronischen Zeitschriften und Repositories.
* Der *International Standard Identifier for Libraries and Related Organisations* (**ISIL**) referenziert Bibliotheken, Archive, Museen und verwandte Einrichtungen.
* Die *Open Researcher and Contributor ID* (**ORCID**) identifiziert Autor\*innen von wissenschaftlichen Publikationen.
* Der *Uniform Resource Locator* (**URL**) dient als Adresse einer digitalen Ressource im Web und wird teilweise gleichzeitig als deren Identifikator eingesetzt.
* Das System der *Uniform Resource Identifier* (**URI**) ermöglicht die Vereinigung verschiedener Identifier-Systeme und bildet die Grundlage von *RDF* und Linked Open Data (*LOD*).

Gemeinsam ist den Identifikatoren, dass sie jeweils eine definierte Syntax haben (z.B. XXXX-XXXY im Falle der ISSN wobei X für eine Ziffer und Y für eine Prüfziffer steht) deren Bestandteile hierarchisch von einer zentralen Instanz festgelegt werden. Nach dem Prinzip des Namensraums kann dabei die Vergabe von Teilen an untergeordnete Organisationen delegiert werden. Beispielsweise werden ISIL für Bibliotheken in Deutschland beginnend mit dem Präfix „DE-“ durch die [ISIL-Agentur an der Staatsbibliothek zu Berlin](https://sigel.staatsbibliothek-berlin.de/) verwaltet.

Völlig dezentrale Identifikatoren gibt es zur Identifizierung von digitalen Objekten nur in Form von Prüfsummen, die sich automatisch aus vorhandenen Daten berechnen lassen (SHA-Summe, IPFS-Adresse, Prüfziffer…).

### Normdaten

Einfache kontrollierte Vokabulare bestehen aus normierten Listen von eindeutigen Benennungen – beispielsweise könnte in einem Gemüse-Vokabular festgelegt sein, dass immer „Karotte“ statt „Möhre“ verwendet werden muss. Wird jeder Eintrag mit einem künstlichen Identifikator versehen, muss die Benennung selbst nicht eindeutig sein. Existiert eine Datenbank zum Nachschlagen dieser IDs, so wird diese auch als *Normdatei* bezeichnet. Ihre Datensätze dienen als Normdaten der eindeutigen Identifizierung von Personen, Organisationen, geographischen oder administrativen Einheiten, Themen oder anderen Entitäten an verschiedenen Stellen. Ein Normdatensatz besteht mindestens aus einem Identifikator und einer Vorzugsbenennung als primärer Name. Oft gibt es weitere identifizierende Merkmale wie alternative Benennungen, Lebensdaten von Personen, Ortsangaben u.ä. sowie Verknüpfungen zwischen verschiedenen Entitäten.

Umfang und Komplexität von Normdateien reichen von einfachen Listen bis zu komplexen Wissensgraphen. Ein prominentes bibliothekarisches Beispiel ist die Gemeinsame Normdatei (*GND*), in der neben Personen auch Körperschaften, Veranstaltungen, Geografika, Werke und Sachschlagwörter miteinander vernetzt sind.

Zur Anreicherung von Daten mit Normdaten mittels *Entity Recognition* (Erkennung von Entitäten in Daten) und *Entity Linking* (Abgleich von Entitäten mit IDs anderer Normdateien) sollte unterschieden werden zwischen:

* Normdateien mit Entitäten wie Personen (ORCID), Publikationen (DOI, ISBN), Sprachen (ISO 639) etc. die sich grundsätzlich eindeutig unterscheiden lassen sowie
* Normdateien, deren abstrakte Entitäten von Kontext und Modellierung abhängen (Klassifikationen, Thesauri…).

Zur Verwaltung von Normdaten gibt es einige Datenformate wie *MARC 21 for Authority Data* und *ISAAR (CPF)*. Als gemeinsamer Nenner auch außerhalb des Bibliotheksbereichs gilt das RDF-basierte *Simple Knowledge Organization System* (*SKOS*), für das mit JSKOS auch eine JSON-Variante existiert. Vor allem kleinere oder speziellere Normdateien (z.B. Codelisten im Rahmen der Katalogisierung) liegen allerdings selten in maschinenlesbarer Form vor.

Das [*Basic Register of Thesauri, Ontologies & Classifications*](https://bartoc.org/) (BARTOC) erfasst Informationen zu Normdateien aller Art, darunter auch Verfahren zum technischen Zugriff.

## Metadatenformate

Neben allgemeinen [Datenformaten](#datenformate) sind für die Bibliotheks-IT vor allem Metadatenformate zur Beschreibung von Dokumenten relevant. Die meisten der im Folgenden beschriebenen Metadatenformate spielen außerhalb von Kultureinrichtungen keine wesentliche Rolle. Für digitale Objekte (*METS/MODS*, *LIDO*, *CDWA*, *EN 15907*, *EAD*… siehe Kapitel *Digitalisierung*) und für Forschungsdaten (DataCite, siehe Kapitel *Forschungsnahe Dienste)* gibt es darüber hinaus spezielle Formate .

### Arten von Metadaten

Folgende Arten von Metadaten können nach ihrer Funktion unterschieden werden:

* **Deskriptive (=beschreibende) Metadaten** zur Identifizierung und inhaltlichen Beschreibung wie Titel, Verfasser, Schlagwörter etc.
* **Administrative (=Verwaltungs-) Metadaten** wie Angaben zu Herkunft, Speicherung, Zugriffsrechte, Verwaltung etc.
* **Strukturelle Metadaten** über den Aufbau von Dokumenten wie die Einteilung in einzelne Kapitel, Abschnitte, eingebundene Medien etc. Die Grenze zwischen digitalen Inhalten und ihrer Struktur ist allerdings mitunter fließend.
* **Technische Metadaten** zu Merkmalen wie Umfang, verwendete Datenformate etc.

Je nach Anwendung gibt es spezielle Metadatenformate oder es werden verschiedene Arten von Beschreibungen in einem Format zusammengefasst.

### Feldbasierte bibliothekarische Metadatenformate

**Machine-Readable Cataloging (MARC)** ist das älteste und noch immer wichtigste Format für den Austausch von Daten zwischen Bibliotheken. Die aktuell relevante Variante ist MARC 21, insbesondere das Format MARC 21 für bibliographische Daten. Neben der binären Kodierung kann MARC 21 auch in XML und JSON kodiert werden. Viele Eigenheiten und Probleme des Formats sind historisch bedingt, eine Alternative konnte sich bislang nicht durchsetzen.

**PICA** ist das von MARC inspirierte Datenformat der Katalogisierungssysteme CBS und LBS (Voß (2022)). Das wichtigste Anwendungsprofil ist das K10plus-Format des BSZ/GBV.

**MAB** und **allegro** sind ebenfalls an MARC angelehnte, feldbasierte Formate aus dem deutschsprachigen Raum, die allerdings nur noch sporadisch verwendet werden.

### Datenmodelle und RDF-Formate

**Dublin Core** bzw. das Dublin Core Metadata Element Set (**DCMES**) hat als kleinster gemeinsamer Nenner der meisten Metadatenstandards die größte Verbreitung. Es besteht aus 15 Basiselementen wie creator, title, date, und description und Erweiterungen mit den DCMI Metadata Terms wie „Alternative Title“, „Date Created“ und „Date Available“.

Die *Functional Requirements for Bibliographic Records* (**FRBR**) sind ein sehr abstraktes Metadatenmodell. Sie beinhalten insbesondere eine Einteilung von bibliographischen Entitäten in die Beschreibungsebenen „work“, „expression“, „manifestation“ und „item“.

Die **BIBFRAME**-Ontologie wurde entwickelt um MARC auf Grundlage von RDF zu ersetzen. Die wesentlichen Elemente sind „work“, „instance“ und „item“ sowie damit verbundene Eigenschaften und Entitätstypen (siehe [Abbildung 6.1](#fig-bibframe)).

**Schema.org** ist eine allgemeine Ontologie für strukturierte Daten in Webseiten.

|  |
| --- |
| Abbildung 6.1: Hauptbestandteile des Datenmodell BIBFRAME |

### Verlagsdaten und Literaturangaben

Die Formate *ONIX*, *JATS*, *BITS* und *CrossRef XML* stammen aus dem Verlagsbereich zur Beschreibung von Zeitschriftenartikeln und Büchern. Sie basieren alle auf XML und sind für Bibliotheken für den Datenimport relevant. Datenformate für Literaturangaben (*BibTeX*, *RIS*, *Endnote*, *CSL-JSON*…) werden dagegen zum Export von Katalogdaten bereitgestellt. Zitationsregeln für Literaturangaben und Ansetzungsregeln von *ISBD* sind dagegen für den Datenaustausch eher unbrauchbar. Learning Object Metadata (*LOM*) dient in verschiedenen lokalen Anpassungen der Beschreibung von Lerneinheiten.

## Datenverarbeitung

Neben der [Erfassung von Daten](#datenerfassung) umfasst das bibliothekarische (Meta)datenmanagement Verfahren zur Konvertierung, Integration und Auswertung von Daten. Diese wesentlichen Schritte werden oft mittels eines ETL-Prozesses zur Übertragung von Daten in ein anderes System vorgenommen. Nach Beschreibung der Bestandteile des ETL-Prozesses wird in diesem Abschnitt auf Werkzeuge und Schnittstellen eingegangen.

### ETL-Prozess

Da sich die IT in- und außerhalb von Bibliotheken über verschiedene Organisationen und Systeme erstreckt, müssen an vielen Stellen Daten von einer oder mehreren Quellen in ein anderes Informationssystem übertragen werden. Der grundsätzliche Prozess der Datenintegration, der Quell- und Zielsysteme verbindet, wird als **ETL-Prozess** bezeichnet. Der Prozess aus drei zentralen Schritten „Extract“, „Transform“ und „Load“ stammt ursprünglich aus dem Bereich des Data Warehousing und findet sich auch in anderen Anwendungsfällen. Im Folgenden wird er am Beispiel der Integration von Metadaten in ein Discovery-System beschrieben. [Abbildung 6.2](#fig-etl) illustriert den generellen ETL-Prozess mit einigen exemplarischen Arbeitsschritten.

|  |
| --- |
| Abbildung 6.2: Beispiel eines ETL-Prozess |

#### Extraktion

Ziel der **Extraktion** (Extract) ist die Auswahl und der Abzug relevanter Daten aus verschiedenen Datenquellen. Hierbei handelt es sich primär um einen technischen Vorgang, dem sogenannten **Harvesting**, welcher automatisiert oder manuell gestartet werden kann. Der Aufwand und die Qualität des Harvesting können je nach Datenquelle sehr unterschiedlich ausfallen. Denkbare Datenquellen sind Dateien, Datenbanken bzw. Datenbankabzüge, [Schnittstellen](#schnittstellen) oder eher unstrukturierte Quellen wie Websites, die zunächst mittels Screenscraping erschlossen werden müssen.

Der Extraktionsvorgang erfolgt bei Bedarf regelmäßig, um die Daten im Zielsystem aktuell zu halten. Mögliche Aktualisierungsintervalle sind:

* **periodisch**, das heißt in zeitlich regelmäßigen Abständen unabhängig von der jeweiligen Aktualisierungen der Daten in den Quellsystemen
* **ereignisgesteuert** immer wenn bestimmte Bedingungen wie zum Beispiel die Änderung von Daten in den Quellsystemen eintreten
* **manuell** beispielsweise, wenn Daten aus Quellsystemen ad hoc importiert werden sollen. Manuelle Aktualisierungen bieten sich vor allem an, wenn der Inhalt der Datenquellen weitestgehend statisch ist, da die dauerhaft verlässliche Aktualisierung im Zielsystem ohne Automatismus nicht garantiert ist.

Der Extraktionsvorgang ist technisch relativ einfach handhabbar, wenn strukturierte [Datenformate](#datenformate) und/oder [Schnittstellen](#schnittstellen) existieren — die wesentlichen Aufwände finden sich dann im nachfolgenden Transformationsschritt. Anders sieht es aus, wenn beispielsweise Daten manuell eingesammelt werden müssen oder Screenscraping notwendig ist. Beim *Screenscraping* müssen aufwändig Extraktionsskripte erstellt werden um Daten aus Webseiten in ein strukturiertes Format zu überführen. Diese Skripte sind zudem sehr fehleranfällig und müssen jedes mal angepasst werden wenn die Betreiber der Datenquelle Veränderungen vornehmen.

Die extrahierten Daten werden in einem sogenannten Arbeitsbereich abgelegt und dort im nächsten Prozessschritt aufbereitet.

#### Transformation

Daten aus verschiedenen Quellsystemen liegen zumeist in unterschiedlichen Formaten mit unterschiedlichen Datenmodellen vor. Neben Unterschieden in der Syntax können gleiche Sachverhalte auch auf semantischer Ebene unterschiedlich beschrieben sein, da die Daten mitunter für abweichende Anwendungsfälle erfasst wurden. So müssen beispielsweise in einem Discovery-System Metadaten zur einfachen Beschreibung so aufbereitet werden, dass sie auch erweiterte Suchstrategien unterstützen.

Ziel der **Transformation** ist es, alle Daten in ein einheitliches Format mit gemeinsamen Datenmodell zu überführen. Dieses Zielformat wird beim ETL-Prozess auch als *Schema* bezeichnet. Die Vereinheitlichung des Schemas (*Mapping*) ist ein wesentlicher Schritt jeder Datenkonvertierung. Zur Minimierung Transformationsaufwand dienen gemeinsame Standards wie *MARC21* als Austauschformat oder die einheitliche Verwendung von RDF-Ontologien.

Über die einfache Konvertierung hinaus sind im Rahmen der Transformation oft weitere Aufbereitungen zur Vereinheitlichung und Verbesserung der Datenqualität notwendig. Beispiele hierfür sind:

* Prüfung und **Vereinheitlichung** der Zeichencodierung auf normalisierten Unicode
* formale **Anpassungen** von Daten wie die Vereinheitlichung von Datumsformaten, Ländercodierungen etc.
* Erkennung und Eliminierung von **Duplikaten**
* Abgleich, Vereinheitlichung und **Konsistenzprüfung** von Aussagen über dieselben Objekte aus verschiedenen Datenquellen
* **Anreicherung** oder Korrektur von Datensätzen mittels Zusatzinformationen aus Normdaten oder anderer zusätzlicher Datenquellen

Der Aufwand der Transformation sollte nicht unterschätzt werden, da die Datenqualität verschiedener Datenquellen stark variieren kann und Qualitätsprobleme oft erst spät entdeckt werden. Zudem können sich Datenquellen und ihre Qualität zwischen Aktualisierungen ändern. Sofern die Datenübernahme nicht nur einmalig stattfinden soll (*Konversion*), ist die Betreuung des Transformationsschrittes eine Daueraufgabe.

Da im Transformationsschritt regelmäßig Massendaten analysiert und modifiziert werden müssen, ist der Einsatz von IT-gestützten [Werkzeugen](#werkzeuge) und Verfahren der [Datenanalyse](#datenanalyse) unerlässlich. Die aus der Analyse gewonnenen Erkenntnisse müssen wiederum kontinuierlich in die Anpassung des Schema-Mappings einfließen, damit der Transformationsprozess nicht ins Stocken gerät.

#### Laden

Auf die Transformation folgt beim **Laden** (load) die Überführung der vereinheitlichten Daten in das Zielsystem – beispielsweise in den Suchindex eines Discovery-Systems. Dabei dürfen nur Datensätze in Produktivsysteme übernommen werden, die den Transformationsschritt erfolgreich durchlaufen haben, während für Test- und Entwicklungssysteme andere Regeln möglich sind.

Das Laden selbst ist ein technisch beherrschbarer Schritt, welcher optimalerweise darauf abzielt, das Zielsystem ohne Ausfallzeiten aktuell zu halten. Bei Änderungen des Schemas muss deshalb besonders darauf geachtet werden, gleichzeitig entsprechende Anpassungen im Zielsystem vorzunehmen. Daneben ist es bei absehbaren Änderungen ratsam, das Schema und die Verarbeitung von Daten im Zielsystem von vornherein flexibel zu gestalten.

#### Umsetzung des ETL-Prozess

Die Umsetzung von Datenkonvertierung und ETL-Prozessen erfolgt im bibliothekarischen Umfeld oft über selbst entwickelte Skripte für das Harvesting und die Transformation. Das entsprechende Know-How ist in der Regel auf wenige Köpfe verteilt und kann auch nicht leicht durch das Hinzuziehen externer Expertise verfügbar gemacht werden.

Es gibt einige kommerzielle ETL-Komplettlösungen mit Data-Warehouse- oder Business-Intelligence-Hintergrund. Angesichts der teils erheblichen Einstiegskosten und zur Vermeidung von *vendor-lock-in* sind für Bibliotheken möglichst einfache und allgemeine [Werkzeuge zur Datenverarbeitung](#werkzeuge) jedoch meist die bessere Wahl. Die Vorteile etablierter ETL-Werkzeuge liegen in Schulungsmöglichkeiten und der Verfügbarkeit externer Expertise. Mit *Catmandu*, *Metafacture* und *OpenRefine* gibt es mehrere Open-Source ETL-Frameworks, deren eingeschränkter Funktionsumfang und Usability durch Anpassungen für bibliothekarische Datenformate und Schnittstellen möglicherweise aufgewogen werden.

Grundsätzlich lassen sich die kontinuierlich anfallenden Aufwände der Transformation und Qualitätssicherung durch ein ETL-Werkzeug nicht vermeiden, sondern nur besser handhabbar machen. Dazu sollten die einzelnen Arbeitsschritte zentral verwaltet und dokumentiert werden, beispielsweise durch ein Versionskontrollsystem. Durch den Fokus auf Usability kann mittels ETL-Werkzeugen bibliothekarisches Personal stärker eingebunden werden. Da hier insbesondere im Bereich der Programmierung die Einstiegshürden niedriger sind, ist es Fachpersonal leichter möglich, direkt Änderungen und Optimierungen am ETL-Prozess vorzunehmen. Bei einer reinen Verankerung von ETL im IT-Bereich sind solche Eingriffe dagegen nur in Zusammenarbeit von IT und Metadaten-Expert\*innen umsetzbar.

In jedem Fall gehen mit der Einführung von ETL-Werkzeugen in die bibliothekarische Arbeit immer auch individuelle Anpassungen im Prozess von Extraktion, Transformation und Laden einher. Dieser Aufwand kann sowohl gegen die Einführung solcher Werkzeuge sprechen als auch dafür, vorhandene „Bastellösungen“ zu evaluieren und zu konsolidieren.

### Werkzeuge

Für die Verarbeitung von Daten im Rahmen bibliothekarischer IT-Systeme werden grundsätzlich die gleichen Werkzeuge verwendet wie außerhalb des Bibliothekswesens, daher wird an dieser Stelle auf eine allgemeine Einführung in die Datenverarbeitung verzichtet. Ganz allgemein sind hier als Werkzeuge

* Mittel zur **Dateiverwaltung** und ein Texteditor unabdingbar,
* allgemeine **Kommandozeilenprogramme** (curl, sort, grep…) sehr zu empfehlen
* und **Programmiersprachen** vor allem für komplexere Aufgaben hilfreich.

Material und Kurse für die praktischen Grundkenntnisse mit Bibliotheksbezug finden sich insbesondere in den Bereichen *Data Librarianship* und *Data Science*.

Werkzeuge für konkrete Datenformate orientieren sich an den zugrunde liegenden [Strukturierungssprachen](#daten): so gibt es beispielsweise eigene Editoren oder Editor-Plugins für XML-Daten und JSON-Daten und entsprechende Kommandozeilentools wie xmlstarlet für XML und jq für JSON. Für tabellarische Daten eignet sich etwa eine Tabellenkalkulation oder das tabellenbasierte Werkzeug *OpenRefine* ([openrefine.org](https://openrefine.org/)).

Für bibliothekarische Datenformate und Schnittstellen gibt es darüber hinaus einige speziellere Werkzeuge:

* Programmierbibliotheken wie *MARC4J* und *YAZ* erleichtern die Datenverarbeitung im Rahmen eigener Programme.
* Anwendungsprogramme wie *WinIBW* und *BibControl* erfordern zwar weniger IT-Kenntnisse, sind dafür aber nur eingeschränkt und/oder nur für sehr spezielle Aufgaben verfügbar.
* Freie Werkzeuge zum Metadatenmanagement wie die Frameworks *Catmandu* ([librecat.org/Catmandu](http://librecat.org/Catmandu/)) und *Metafacture* ([metafacture.org](https://metafacture.org/)).

Für einzelne Anwendungen und Formate gibt es einige weitere Werkzeuge und es kann sich lohnen, solche Werkzeuge selbst zu entwickeln und als Open Source zur Verfügung zu stellen. Für das PICA-Format sind solche Programme in der *Einführung in die Verarbeitung von PICA-Daten* (Voß (2022)) aufgeführt.

### Schnittstellen

Eine API (Application Programming Interface, auch Programmierschnittstelle) ist eine definierte Methode zur Abfrage und/oder Änderung von Daten in einem Informationssystem. Wie die Daten innerhalb des Systems verwaltet werden, ist dabei nebensächlich. Dieses Prinzip ermöglicht die Kombination unterschiedlicher Softwarekomponenten. Wenn möglich, sollten produkt-unabhängige, offen dokumentierte APIs verwendet werden. Im Bibliotheksbereich sind insbesondere folgende APIs relevant:

* **Z39.50** wurde vor Erfindung des Web zur Suche in Bibliotheksdatenbanken entwickelt. Nachfolger ist das XML-basierte *Search/Retrieve via URL* (**SRU**) mit der zugehörigen Abfragesprache *Contextual Query Language* (**CQL**).
* Das *Open Archives Initiative Protocol for Metadata Harvesting* (**OAI-PMH**) dient dem Abruf von Metadaten aus Repositorien. Die Daten können nach Datum und Teilmengen gefiltert werden.
* Das *NISO Circulation Interchange Protocol* (**NCIP**) und das *Standard Interchange Protocol* (**SIP2**) sind APIs zur Ausleihe und Fernleihe. Sie werden intern zwischen Bibliotheken und zur Anbindung von Verbuchungsautomaten eingesetzt.
* Die *Patrons Account Information API* (**PAIA**) ist eine offene Schnittstelle zum Zugriff auf Ausleihkonten.
* Die *Document Availability Information API* (**DAIA**) ist eine offene Schnittstelle zur Abfrage der Verfügbarkeit von Medien.
* Mit **unAPI** können einzelne Datensätze in verschiedenen Formaten abgerufen werden.
* Die **Reconciliation Service API** ermöglicht den Abgleich mit Normdaten zur eindeutigen Referenzierung (siehe Abschnitt [Identifikatoren und Normdaten](#identifikatoren-und-normdaten)).
* Verschiedene Schnittstellen zur Authentifizierung und Autorisierung wie *LDAP*, *Shibboleth* und *OAuth*.

Darüber hinaus bieten die meisten Anwendungen eigene Schnittstellen, zum Beispiel die Solr-API der Suchplattform *Apache Solr*. Weitere Schnittstellen spielen eher im Bereich Digitalisierung und Forschungsnahe Dienste eine Rolle, beispielsweise die APIs des International Image Interoperability Framework (*IIIF*).

## Datenanalyse

Im Gegensatz zu physischen Objekten sind Daten ihre Beschaffenheit nicht direkt anzusehen. Lediglich der Umfang von Daten in Bytes und ggf. die Anzahl von Dateien und Datensätzen kann einen ersten Anhaltspunkt liefern. Weitere Einschätzungen, insbesondere darüber, ob Daten vollständig oder fehlerhaft sind, setzen eine konkrete Analyse der Daten voraus. Dies beinhaltet auch die Visualisierung von Daten zur Exploration, Kommunikation und Diskussion.

Die Auswertung von Daten ist nicht nur für das Qualitätsmanagement relevant (Voß (2021)), beispielsweise um im Rahmen des ETL-Prozesses Verteilungen und Ausreißer zu erkennen, sondern auch um aus Daten weitere Erkenntnisse zu gewinnen. So können beispielsweise Ausleihzahlen nach Medien gruppiert für die Bestandsplanung eingesetzt werden. Voraussetzung dafür ist, dass Daten überhaupt vorliegen, sowie Mittel und Kenntnisse zu ihrer Auswertung. Bei fehlenden oder zu umfangreichen Daten können Stichproben erhoben werden, wobei auf Zufälligkeit der Stichprobe und Konfidenzintervall des Ergebnisses geachtet werden muss.

Neben rudimentären Statistik-Kenntnissen helfen bei der Datenanalyse Werkzeuge wie die [im vorigen Abschnitt](#werkzeuge) beschriebenen Mittel zur Datenverarbeitung. Für allgemeine Analysen eignet sich vor allem eine Tabellenkalkulation. Für komplexere Analysen gibt es spezielle Statistik-Programme und Programmiersprachen wie *SAS* und *R*. Für explorative Analysen und um bei geänderter Datenlage automatisch aktuelle Ergebnisse zu bekommen, bieten sich interaktive Umgebungen wie *Jupyter Notebooks* oder *Observable* an*.* Für spezielle, wiederkehrende Analysen und Aufbereitungen kann es auch sinnvoll sein, eigene Anwendungen zu entwickeln bzw. entwickeln zu lassen. Beispiele hierfür sind *BibControl*, das *Metadata Quality Assessment Framework* (*MQA*), die Deutsche Bibliotheksstatistik sowie Statistikfunktionen als Teil anderer Programme (zum Beispiel [Statistik und Reporting](#statistik-und-reporting) als Teil des BMS).

# Literaturverzeichnis

Hier nur die explizit im Handbuch referenzierten Publikationen. Die vollständige Bibliographie wird [in einer gemeinsamen Zotero-Gruppe](https://www.zotero.org/groups/4673379/it_in_bibliotheken) verwaltet.

Borgman, Christine L. 1997. „From Acting Locally to Thinking Globally: A Brief History of Library Automation“. *The Library Quarterly: Information, Community, Policy* 67 (3): 215–49. <https://www.jstor.org/stable/40039721>.

Breeding, Marshall. 2022. „How to Secure Library Systems From Malware, Ransomware, and Other Cyberthreats“. 2022. <https://www.infotoday.com/cilmag/jan22/Breeding--How-to-Secure-Library-Systems-From-Malware-Ransomware-and-Other-Cyberthreats.shtml>.

Breeding, Marshall. o. J. „Library Technology Industry Mergers and Acquisitions. Library Technology Guides“. Zugegriffen 28. April 2022. <http://librarytechnology.org/mergers/>.

Deutsches Institut für Normung e. V. (DIN). 2020. „DIN EN ISO 9241-110 Ergonomie der Mensch-System-Interaktion - Teil 110: Interaktionsprinzipien (ISO 9241-110:2020)“. <https://www.din.de/de/mitwirken/normenausschuesse/naerg/veroeffentlichungen/wdc-beuth:din21:320862700>.

Gesetzliche, Deutsche, und Unfallversicherung e.V. (DGUV). 2019. „Bildschirm- und Büroarbeitsplätze: Leitfaden für die Gestaltung“. <https://publikationen.dguv.de/widgets/pdf/download/article/409>.

Gould, J. D., und C. Lewis. 1987. „Designing for usability: Key principles and what designers think“. In *Human-computer interaction: a multidisciplinary approach*, 528–39. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

„Government Design Principles. GOV.UK“. 2012. 2012. <https://www.gov.uk/guidance/government-design-principles>.

Hanson, Cody. 2015. „Opinion: Libraries are Software“. 2015. <https://www.codyh.com/writing/software.html>.

Jakob Nielsen. 2000. „Why You Only Need to Test with 5 Users. Nielsen Norman Group“. 2000. <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.

Kling, Rob, und Susan Leigh Star. 1998. „Human centered systems in the perspective of organizational and social informatics“. *ACM SIGCAS Computers and Society* 28 (1): 22–29. <https://doi.org/10.1145/277351.277356>.

Kluge, Matthias. 2022. *Anbieter von Bibliothekssoftware*. Landesfachstelle für das öffentliche Bibliothekswesen. <https://www.oebib.de/fileadmin/redaktion/2_bau_einrichtung_it/2_3_it_internet/Bibliothekssoftware/2_edv_anbieter_bibliothekssoftware_auswahl_2022_kl.pdf>.

Matthews, Joseph R., und Carson Block. 2020. *Library information systems*. Second edition. Library and information science text series. Santa Barbara, California: Libraries Unlimited.

Schweitzer, Roswitha. 2016. „Anforderungen an ein Bibliothekssystem der neuen Generation - der Kriterienkatalog von hbz und VZG“. Köln. <https://docplayer.org/61296444-Anforderungen-an-ein-bibliothekssystem-der-neuen-generation.html>.

Shneiderman, Ben, und Catherine Plaisant. 2005. *Designing the user interface. Strategies for effective human-computer interaction*. 4th Aufl. Pearson.

Siems, Renke. 2022. „Das Lesen der Anderen: Die Auswirkungen von User Tracking auf Bibliotheken“. *o-bib. Das offene Bibliotheksjournal / Herausgeber VDB* 9 (1): 1–25. <https://doi.org/10.5282/o-bib/5797>.

Voß, Jakob. 2021. „Datenqualität als Grundlage qualitativer Inhaltserschließung“. In *Qualität in der Inhaltserschließung*, 167–76. De Gruyter Saur. <https://doi.org/10.1515/9783110691597-010>.

———. 2022. „Einführung in die Verarbeitung von PICA-Daten“. 2022. <https://pro4bib.github.io/pica/>.

# Glossar

[API](https://de.wikipedia.org/wiki/Programmierschnittstelle)

Application Programming Interface, Programmierschnittstelle

BMS

Bibliotheksmanagementsystem, gebräuchlich auch als LMS (Library Management System)

[COUNTER](https://www.projectcounter.org/)

Counting Online Usage of NeTworked Electronic Resources

[Discovery-System](https://de.wikipedia.org/wiki/Discovery-System)

Auf Suchmaschinentechnologie beruhende Systeme zur Suche in großen (auch externen) Datenbeständen

[EDIFACT](https://de.wikipedia.org/wiki/EDIFACT)

Electronic Data Interchange for Administration, Commerce and Transport

[ERP](https://de.wikipedia.org/wiki/Enterprise-Resource-Planning)

Enterprise-Ressource-Planning, Ressourcen wie z.B. Finanzen unternehmerisch überwachen und planen

FID

Fachinformationsdienst

[GOKb](https://gokb.org/)

Global Open Knowledgebase, eine Open Data Plattform

ID

Eindeutige Referenz auf einen Datensatz (Identifikator)

[IDM](https://en.wikipedia.org/wiki/Identity_management)

Identity Management, das Speichern von Metadaten zu Personen

[KBART](https://www.niso.org/standards-committees/kbart)

Knowledgebases and related tools, Datenformat zum Transfer von Metadaten

[Lock-In-Effekt](https://de.wikipedia.org/wiki/Lock-in-Effekt)

Kundenbindung durch hohen Wechselaufwand

[MARC](https://de.wikipedia.org/wiki/Machine-Readable_Cataloging)

MAchine-Readable Cataloging, ältestes und noch immer wichtigstes bibliothekarisches Austauschformat

Normdaten

Datensätze zur eindeutigen Identifizierung und Referenzierung von Entitäten wie Personen, Organisationen, Themen…

[OPAC](https://de.wikipedia.org/wiki/OPAC)

Online Public Access Catalogue, Katalog einer Bibliothek

[Persona](https://de.wikipedia.org/wiki/Persona_(Mensch-Computer-Interaktion))

„Maske“, ein Modell zur Beschreibung eines Anforderungsszenario

[SaaS](https://de.wikipedia.org/wiki/Software_as_a_Service)

Software as a Service, Software und Hardware bei externem Dienstleister

[Usability](https://de.wikipedia.org/wiki/Gebrauchstauglichkeit_(Produkt))

Gebrauchstauglichkeit

[User Experience](https://de.wikipedia.org/wiki/User_Experience)

Nutzererfahrung

[VR](https://de.wikipedia.org/wiki/Virtuelle_Realität)

Virtual Reality, (computer-generierte) virtuelle Umgebung

[WCAG](https://www.w3.org/WAI/standards-guidelines/wcag/)

Web Content Accessibilty Guidelines

Z39.50

Protokoll zur Abfrage von bibliografischen Daten

# Hinweise zur Mitarbeit

Dieses Handbuch wird in einem offenen Prozess erarbeitet und verbessert. Mitarbeit ist dabei grundsätzlich sehr willkommen und in verschiedener Form möglich. Voraussetzung ist ein Verständnis des [Arbeitsablauf](#arbeitsablauf), Einhaltung des [Styleguide](#styleguide) und die freie [Lizensierung](#lizenz) der eigenen Beiträge. Das Verfahren zur Aktualisierung des Handbuchs ist unter [Technik](#technik) beschrieben.

## Arbeitsablauf

Der aktuelle Workflow beinhaltet folgende Schritte:

Schreiben

Neue Themen werden durch Gruppen im Rahmen von Book Sprints erstellt. Als Werkzeug zum freien Schreiben dient dabei beispielsweise Google Docs.

Lektorat

Fertige Kapitel werden einmalig nach Markdown konvertiert und im [git-Repository](#technik) des Handbuch abgelegt. Anschließend werden daraus *DOCX*-Dateien erstellt (Word) und in einem [Google Drive Verzeichnis](https://drive.google.com/drive/folders/1JMBLJlk71JqQMQY7j_uXwV47fX8NA_N2) zum Korrekturlesen und Kommentieren bereitgestellt. Änderungen können auch [mittels GitHub-Issues](https://github.com/pro4bib/handbuch-it-in-bibliotheken/issues) oder per Hinweis an [die Autor\*innen](#autorinnen) erfolgen.

Redaktion

Änderungen an bestehenden Kapiteln und an der Gesamtstruktur des Buches können direkt im git-Repository vorgenommen werden. Alternativ muss jemand Änderungsvorschläge aus den Kapitel-Kopien bei Google Drive einarbeiten und die DOCX-Datei aktualisieren.

Publikation

Änderungen an der Markdown-Dateien im git-Repository führen dazu, dass das Buch mittels [quarto](https://quarto.org/) in HTML und anderen Formaten aus den Quellen zusammengebaut wird.

## Styleguide

Dieser Styleguide soll eine einheitliche Form trotz [unterschiedlicher Autor\*innen](#mitarbeit-autorinnen) gewährleisten. Dazu gibt es Hinweise zu [Zielgruppe](#zielgruppe), [Stil und Aktualität](#schreibstil), [Schreibweise](#schreibweise) und [Struktur](#struktur) sowie Vorgaben zu besonderen Inhalten wie [Bildern und anderen Medien](#mitarbeit-medien), [Literaturverzeichnis](#mitarbeit-literatur) und [Glossar](#mitarbeit-glossar).

### Zielgruppe

Zur Klärung der Zielgruppe dieses Handbuchs wurden einige sogenannte *Personas* definiert:

* **Janine** Buchinger: Janine leitet die Stadtbibliothek in einer Stadt mit 250.000 Einwohnern. Die Bibliothek besteht aus einer Zentrale und zwei Zweigstellen. Mit den Schulbüchereien besteht eine Kooperation für fachliche Beratung und gemeinsame Aktivitäten bei der Informationskompetenz-Vermittlung.
* Dr. **Tillmann** Schuppe: Tillmann ist Leiter einer Fachhochschulbibliothek mit 500.000 Medieneinheiten. Die Bibliothek gehört einem Bibliotheksverbund an. Die Bibliothek plant einen Neubau, der gemeinsam mit dem Rechen- und Medienzentrum bezogen werden soll.
* **Magda** Olsowski: Magda ist studierte Informatikerin und leitet die Gruppe Forschungsdatenmanagement an einer großen Universitätsbibliothek. Sie hat keine bibliothekarische Vorbildung.
* **Alicia** Meyer: Alicia studiert Bibliotheksmanagement und plant eine Masterarbeit, in der sie die Implementierungsprozesse von Software analysieren möchte.
* **Robert** Pohlmann: Robert leitet die IT-Abteilung einer mittelgroßen Universitätsbibliothek und ist nebenberuflich Lehrbeauftragter für einen bibliothekarischen Studiengang.

### Stil und Aktualität

* **Journalistische oder enzyklopädische Neutralität sind nicht oberstes Prinzip dieses Buches.** Es soll vielmehr fundiert und praxisorientiert informieren und beraten und darf dabei auch parteiisch sein.
* **Dieses Buch ist keine wissenschaftliche Forschungsveröffentlichung.** Nicht jede Aussage muss mit einer Quelle belegt werden. Für die Anwendung in der Hochschullehre reicht es, wenn das Buch zentrale Aussagen belegt bzw. auf die wichtigsten aktuellen Studien verweist und somit auch Tipps zur weiterführenden Lektüre bietet.
* **Das Buch sollte in 2-5 Jahren noch aktuell und verständlich sein, aber nicht mehr unbedingt in 10 Jahren.**
* **Das Buch soll als Nachschlagewerk dienen, das nicht vollständig durchgelesen werden muss.** Dabei helfen Redundanz und Querverweise (siehe [Hinweise zur Struktur](#struktur).

### Struktur

#### Struktur des Texts

* Wir verwenden **kurze, unverschachtelte Sätze**.
* Wir erzeugen **Sinnabschnitte**, die möglichst für sich stehend verständlich sind.
* Wir schreiben **stark strukturiert**, also
  + mit vielen **Zwischenüberschriften**, bis maximal zur vierten Gliederungsebene,
  + wo es inhaltlich passt, in stichpunktartigen **Listen** und
  + mit Hervorhebung **wichtiger Begriff durch Fettdruck** als Gliederungshilfe.
* Wir verwenden [**Infoboxen**](https://quarto.org/docs/authoring/callouts.html), die auch unabhängig vom übrigen Text lesbar sind.
* Wir liefern **wichtige Informationen zusätzlich** zum Text in Form von Bildern, Tabellen, Listen, Infoboxen und/oder Zusammenfassungen.

#### Struktur der Hauptkapitel

Das Handbuch behandelt [aufgeteilt in Hauptkapitel](#inhalt) die wesentlichen Themen rund um IT in Bibliotheken.

* Kapitel haben normalerweise einen Umfang von 3.000 bis 4.000 Wörtern. Deutlich längere Kapitel sind darauf zu prüfen, ob sie sich in mehrere Kapitel trennen lassen, und wenn das nicht möglich ist, müssen sie sorgfältig in Unterkapitel aufgeteilt werden.
* Jedes Kapitel beginnt mit einer **Kurzfassung** als Infobox gefolgt von einer **Einleitung** und endet mit einem Abschnitt **Zusammenfassung und Ausblick**.
* Jedes Kapitel beinhaltet ein aussagekräftiges Metadatenfeld description [für Suchmaschinen](https://developers.google.com/search/docs/advanced/appearance/snippet) (maximal etwa 158 Zeichen)

### Schreibweise, Fachbegriffe und Verweise

* Wir verwenden im gesamten Buch gendergerechte Schreibweise mit Sternchen (\*). In Markdown ist es sicherer dem Sternchen einen Backslash voranzustellen, z.B. Autor\\*innen.
* Eine Schreibweise für häufig verwendete Fachbegriffe sollte quer durch das Buch eingehalten werden, so z.B. BMS für Bibliotheksmanagementsysteme
* Fachbegriffe (z.B. Bibliotheksverbund) werden dort verwendet, wo sie wiederholt relevant sind, und werden bei ihrer ersten Erwähnung definiert. Die eingeführten Fachbegriffe müssen in einem Glossar für das gesamte Buch gebündelt werden.
* Wir vermeiden IT-Jargon.
* Quellen sollten nur dann genannt werden wenn in der jeweiligen Textpassage auch wirklich paraphrasiert oder wörtlich zitiert wird.

#### Typografie

* Abkürzungen werden durch geschützte Leerzeichen getrennt (z. B.)
* Gerade Anführungszeichen "..." werden automatisch durch die Deutschland und Österreich übliche Anführungszeichen („…“) ersetzt.

Externe Links, die nur auf Anbieter oder andere Websites verweisen, werden inline verlinkt.

* Links, die auf später entstehende Kapitel verweisen, werden durch eckige Klammern kenntlich gemacht.
* *Kursive* Hervorhebung sollte nur für Namen und [Glossareinträge](#mitarbeit-glossar) verwendet werden.
* Wir verzichten auf Fußnoten.

### Bilder und andere Medien

* Bilder und andere Mediendateien kommen in das Verzeichnis media im [git-Repository](#technik). Alternativ können sie von externen Quellen per URL eingebunden werden wenn die Quelle voraussichtlich dauerhaft verfügbar ist.
* Bilder sollten möglichst als Vektorgrafik (SVG) bereitgestellt werden.
* Bitte nutzt sprechende Dateinamen!

### Literaturverzeichnis

Die zitierte und weiterführende Literatur wird in einer Zotero-Gruppe unter <https://www.zotero.org/groups/4673379/it_in_bibliotheken> verwaltet. Der BibLaTex-Export dieser Bibliographie wird mit Aufruf von make refs von dort heruntergeladen und unter references.bib gespeichert. Diese Datei sollte also nicht direkt bearbeitet werden! Innerhalb des Markdown-Quelltext kann mittels [Pandoc-Citation Syntax](https://quarto.org/docs/authoring/footnotes-and-citations.html#sec-citations) und dem jeweiligen Citekey aus references.bib auf Literatur verwiesen werden.

### Glossar

Das Glossar in der Datei glossar.yml enthält erklärungswürdige Begriffe mit Kurzbeschreibung und optionalem Link auf eine weiterführende Quelle (meist Wikipedia). Es werden \_keine\_ Firmennamen in das Glossar aufgenommen, auch wenn sie Akronyme sind. Die Glossarbegriffe werden in den Textdateien zur Hervorhebung *kursiv* gesetzt (in Markdown so ein \*Fachbegriff\*). Bei Erzeugung der HTML-Version des Handbuchs wird die Hervorhebung in einen Tooltip umgewandelt.

### Autor\*innen-Verzeichnis

Wenn Du etwas beigetragen hast und möchtest, dass Du im [Verzeichnis der Autor\*innen](#autorinnen) auftauchst, trage Dich in der YAML-Datei [contributors.yml](https://github.com/pro4bib/handbuch-it-in-bibliotheken/blob/main/contributors.yml) ein. Die Einträge sollten nach Nachname sortiert werden. Die Felder email, position und orcid sind optional.

## Technik

Die Master-Version des Handbuch liegt in einem git-Repository unter <https://github.com/pro4bib/handbuch-it-in-bibliotheken>. Die Ergebnisdateien werden automatisch via GitHub und einen Server der *VZG* aktualisiert, so dass unter <https://it-in-bibliotheken.de/> immer der aktuellste Stand einsehbar sein sollte.

### Verzeichnisstruktur

Die Markdown-Dateien im Wurzelverzeichnis (\*.md) sind die Masterdateien.

* metadata.yml bibliographische Metadaten (Titel, Abstract…)
* \_quarto.yml zentrale Konfigurationsdatei zur Anpassung der [Konvertierung mit Quarto](#konvertierung)
* <contributors.yml> [Autor\*innen-Verzeichnis](#autorinnen-verzeichnis)
* references.bib [Literaturverzeichnis](#literaturverzeichnis) (bitte nicht direkt bearbeiten!)

Weitere Unterverzeichnisse:

* media/ [Bilder und andere Medien](#bilder-und-andere-medien)

Die Dateien in folgenden Verzeichnissen sollen nicht per Hand geändert werden:

* docs/ aus den Masterdateien mit quarto erzeugte Publikation
* \_gdrive/ von bzw. nach Google-Drive importierte bzw. exportierte Kapitel (siehe [README.md](_gdrive/README.md))

### Konvertierung

Zur Anpassung der Konvertierung des Handbuchs mit [Quarto](https://quarto.org) muss das Repository lokale geklont und Quarto installiert werden. Für die DOCX-Ausgabe muss außerdem rsvg-convert installiert werden (Paket librsvg2-bin bzw. libsrvg).

Die Aufrufe sind zur Vereinfachung in [Makefile](../Makefile) zusammengefasst:

* make preview konvertiert das Handbuch nach HTML und startet einen Webserver mit Vorschau unter <http://localhost:15745/> (PLZ von Wildau). Die HTML-Ansicht wird automatisch aktualisiert wenn die Quelldateien lokale geändert werden.
* make build konvertiert das Handbuch in alle konfigurierten Formate und legt die Ergebnisse im Verzeichnis \_book ab. Dieser Schritt wird auch automatisch nach jedem Push auf GitHub ausgeführt.
* make all ruft make build und docx auf und kopiert die DOCX-Dateien ins Publikationsverzeichnis \_book.
* make html erzeugt nur HTML in \_book.
* make docx erzeugt nur DOCX in \_book.
* make refs aktualisiert das [Literaturverzeichnis](#literaturverzeichnis) in references.bib von Zotero.

## Lizenz

Alle Beiträge werden unter der Lizenz *Creative Commons Namensnennung 3.0 Deutschland* ([CC BY 3.0 DE](https://creativecommons.org/licenses/by/3.0/de/)) veröffentlicht. Für Abbildungen kann auch eine CC-BY-Lizenz (kein -NC oder -ND) verwendet werden.