

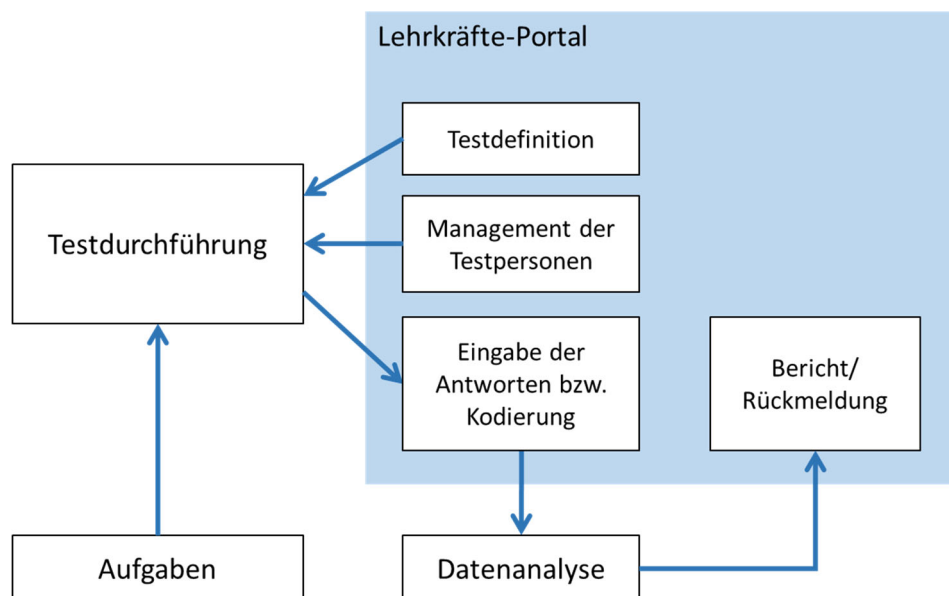
# IQB-Programmierungen zum computerbasierten Testen

## Leitfaden zur Integration Player/Testcenter

27.3.2023

Beginnend mit dem Projekt „TBA-Machbarkeitsstudie“ 2018 beauftragen die Länder der Bundesrepublik Deutschland das Institut zur Qualitätsentwicklung im Bildungswesen (IQB) mit der Entwicklung technischer Systeme zur computergestützten Durchführung von Lernstandserhebungen. Für die landesweiten Vergleichsarbeiten (s. KMK-Strategie zum Bildungsmonitoring VERA3 und VERA8) wurde ein Stufenplan für die Umstellung der bisher papierbasierten Durchführung beschlossen.

Die Software wird den [Servicestandards des Onlinezugangsgesetzes](#) folgend als Open Source entwickelt (Lizenz: MIT). Installation und Betrieb liegt komplett in Verantwortung der Länder, d. h. das IQB stellt keine Infrastruktur für die Durchführung der Lernstandserhebungen bereit. Ein wichtiger Grundsatz bei der Entwicklung ist daher die Integrierbarkeit der IQB-Komponenten in existierende oder in Eigenregie der Länder entstehende technische Lösungen. Da die Programmierungen des IQB nur die Durchführung einer Lernstandserhebung abdecken, müssen Funktionen wie Verwaltung der Testpersonen, Auswertung und Präsentation der Ergebnisse weiterhin über die Länder bzw. die mit der Durchführung beauftragten Einrichtungen bereitgestellt werden. Die untere Grafik zeigt eine übliche Konstellation von Anwendungen:



Dieser Text soll helfen, die Integration zu planen und umzusetzen. Er stellt eine Zusammenfassung von diversen online verfügbaren Dokumentationen dar und richtet sich an Auftraggeber ebenso wie Auftragnehmer für IT-Entwicklungsleistungen. Der Text bezieht sich dabei auf den aktuellen Entwicklungs- und Planungsstand und unterliegt daher einer regelmäßigen Revision. Die Arbeitsplanung muss stets unter Vorbehalt gesehen werden bezüglich Änderungen der verfügbaren Ressourcen und der Prioritäten der Auftraggeber.

# Überblick Architektur

Grundsätzlich werden die Programmierungen des IQB als Webanwendungen erstellt. Sie gliedern sich entsprechend ihrer Funktionen:

- Testcenter: Schülerinnen und Schüler (die „Testpersonen“) rufen eine Internetseite auf, geben Login-Daten ein, laden und starten den Test. Im Standardfall wird von einem Endgerät mit Browser (Desktop-Computer, Tablet, Laptop) über das öffentliche Internet auf einen Webserver zugegriffen. Sollte die Internetverbindung zu schlecht oder unsicher sein, kann auch ein zum Test mitgebrachter Laptop die Funktion des Webserver übernehmen.
- Studio: Mit dieser Anwendung werden die Seiten des Tests („Aufgaben“) erstellt.
- Kodierbox: Diese Anwendung dient der Aufbereitung der Antworten für die Analyse.

Das Format der erstellten Aufgaben ist nicht dokumentiert, um eine maximale Flexibilität zu erreichen. Statt dessen werden die Code-Module, die für die Erstellung und Anzeige der Aufgaben benötigt werden („Player“, „Editor“ usw.), als Plugins mit einer standardisierten Schnittstelle bereitgestellt („Verona“). Wenn Anwendungen einmalig diese Schnittstelle implementieren, können nachfolgend immer wieder neue Aufgaben in Tests verwendet werden, selbst wenn sich Interaktionskonzepte für die Beantwortung grundlegend ändern.

Die Anwendungen „Studio“ und „Kodierbox“ werden vom IQB vor allem für die interne Nutzung entwickelt. Die Anwendung „Testcenter“ hingegen wird so programmiert und veröffentlicht, dass sie auch durch die Länder bzw. mit der Testdurchführung beauftragte Einrichtungen nutzbar ist. Änderungen an der Backend-Schnittstelle (API) sind selten und werden ausführlich dokumentiert.

## Option Player

Um eine vom IQB oder einer anderen Quelle gelieferten Verona-Aufgabe zu präsentieren („abzuspielen“), ist ein Player nötig. Es handelt sich dabei um eine Html-Datei mit eingebettetem JavaScript-Code, die in ein iframe-Element (Attribut doc bzw. srcdoc) geladen wird und über `postMessage()` mit dem Host kommuniziert. Aus der Sicht der Testperson handelt es sich um einen Bildschirmbereich.

Ein Test bzw. eine Befragung besteht stets aus Units – entweder Formulare bzw. Interaktionselemente mit dem Ziel, Informationen über die Testperson zu erhalten, oder es handelt sich um Seiten, die nur angezeigt werden sollen (Begrüßung, allgemeine Instruktionen, Beispielaufgaben usw.). Der Ablauf bei einer Navigation ist stets gleich:

1. Laden des Players in das iframe-Element → Player meldet Bereitschaft
2. Player bekommt Unit-Definition und ggf. vorherige Antworten → Player präsentiert Inhalte
3. Player meldet jede Aktion der Testperson an den Host → Host speichert

Die Kommunikation ist standardisiert: „Verona-Spezifikation“. Wenn also eine Webanwendung diese Schnittstelle implementiert, können Player und damit Aufgaben aus dem Verona-Bereich genutzt werden. Folgende weitere Funktionen sind in der Player-Schnittstelle vorgesehen:

- Der Player meldet, wenn er den Focus des Computersystems erhält und wenn er ihn verliert. Dies kann man nutzen, um unerwünschtes Verhalten der Testperson (anderes Browserfenster öffnen) feststellen möchte.

- Der Player kann die Navigation zu einer anderen Unit auslösen (bzw. Vorwärts-/Rückwärtsblättern). Wenn man entsprechende Navigationsschalter auf der Unit-Fläche definiert, kann man den gesamten Bildschirm für die Darstellung der Unit nutzen.
- Der Host kann den Player anweisen, auf eine bestimmte Unterseite zu wechseln, sollte die Unit aus mehreren Seiten bestehen.
- Der Player kann dem Host melden, ob die Präsentation der Unitinhalte (Seiten, Audios usw.) vollständig abgeschlossen ist. Davon kann der Host die Möglichkeit des Weiterblätterns abhängig machen.
- Der Player kann dem Host melden, ob die Beantwortung der Formularelemente der Unit vollständig (und valide) abgeschlossen ist. Davon kann der Host die Möglichkeit des Weiterblätterns abhängig machen.

Das IQB hat eine Reihe von Playern programmiert, die spezifische Anforderungen erfüllen. Der Hauptplayer „Aspect“ bietet die Grundlage für die VERA-Aufgaben.

## Option IQB-Testcenter

### Installation

Das IQB-Testcenter ist eine Webanwendung, die über Docker installiert wird. Alle notwendigen Teile der Anwendung (Frontend, Backend, Datenbank usw.) sind als separate Container vorbereitet. Während des Installationsprozesses werden die Images von [hub.docker.com](https://hub.docker.com) auf den Server geladen und gestartet. Die Container sind (Stand März 2023) nicht unabhängig und können daher nicht skaliert bzw. in Lastverteilungsszenarien (Kubernetes, Docker Swarm u. ä.) eingesetzt werden.

Die Installationsskripte des IQB sehen übliche Linux-Cloudserver vor, die lediglich Docker mit dem Compose-Plugin sowie Make installiert haben müssen. Sollten spezielle Instrumente zum Management der Container zum Einsatz kommen (z. B. Portainer), kann auch manuell über Konfigurationsdateien eingegriffen werden.

### Betrieb

Nach der Installation werden Accounts für die System-Administration angelegt. Diese System-Accounts können

- das Erscheinungsbild der Anwendung ändern (Hintergrundfarbe, Logo),
- den Einführungstext der Anwendung, Impressum und Datenschutztext setzen,
- auffällige Nachrichten auf dem Anmeldebildschirm platzieren, z. B. um ein Wartungsfenster anzukündigen,
- Arbeitsbereiche für die Testungen anlegen und
- Nutzer-Accounts anlegen, die auf die Arbeitsbereiche zugreifen dürfen (RO oder RW)

Die Durchführung einer Studie bzw. einer Befragung gliedert sich in folgende Schritte:

#### *1. Vorbereitung: Testmaterial hochladen*

Das Material und die Konfigurationen für die Durchführung werden in Form von Dateien in das System geladen. Es gibt keine Dialoge im Testcenter, über die man Parameter des Tests oder die Logins verändert, sondern die gesamte Steuerung erfolgt über Informationen, die in den hochgeladenen Dateien gespeichert sind. Eine Änderung der Testdurchführung ist daher nur möglich, indem eine berechtigte Person Dateien aus einem Arbeitsbereich entfernt bzw. austauscht.

Folgende Typen von Dateien werden unterschieden:

- Units: Über diese Dateien werden die einzelnen Aufgaben bzw. Seiten eines Test definiert. Es muss jeweils eine XML-Datei mit Metadaten hochgeladen werden und mitunter zusätzlich eine Inhaltsdatei, z. B. JSON. Für die Darstellung von Units ist noch mindestens ein Code-Modul „Player“ notwendig sowie bei Bedarf sog. Pakete, die der Player ggf. während der Darstellung einer Unit nachlädt.
- Booklets: In diesen XML-Dateien ist der Testablauf definiert, also die Abfolge der Units und ggf. Bedingungen für die Präsentation (z. B. Zeitbeschränkungen bei der Beantwortung, Benennung von Schaltern usw.).
- Testtaker: In diesen XML-Dateien werden die Zugangsdaten für die Testpersonen hinterlegt sowie die Booklets aufgeführt, die jeweils der Person vorgelegt werden sollen.

Die Dateien werden beim Hochladen validiert und ggf. abgewiesen. Dies bezieht sich zunächst auf die Syntax der Daten (XML). Da aber auch Abhängigkeiten zwischen den Dateien bestehen, werden z. B. Booklets abgewiesen, die auf nicht vorhandene Units verweisen. Das bedeutet, dass mitunter die Reihenfolge des Hochladens relevant ist. Außerdem verhindert das System, dass Anmeldenamen auf einem Server doppelt definiert werden (also übergreifend über alle Arbeitsbereiche).

## *2. Durchführung*

Über die in der Testtaker-Datei definierten Logins erhalten die Testpersonen Zugriff auf Booklets. Mit Klick auf ein Booklet startet der Ladevorgang aller Testinhalte. Je nach Konfiguration des Tests startet die Beantwortung nach dem Laden oder bereits früher und der Ladevorgang wird im Hintergrund fortgesetzt. Technisch bedeutet „Beantwortung“ in diesem Zusammenhang, dass Zustandsänderungen sofort an den Server gemeldet werden zum Speichern. Bei einem Neustart des Tests findet die Testperson die vorher gegebenen Antworten wieder vor.

Die Testleitung vor Ort kann eine sog. Testleitungskonsole aufrufen. Diese bietet eine Sicht auf den Bearbeitungsstand aller Testpersonen einer Testgruppe (Klasse) und erlaubt Manipulationen des Testablaufes (alle pausieren, alle in einen bestimmten Testblock zwingen u. ä.). Außerdem kann über den Zugang zum Arbeitsbereich der Fortgang der gesamten Studie beobachtet werden.

## *3. Nachbereitung: Daten herunterladen*

Die Daten, die nach einem Test bzw. einer Befragung zum Download bereitstehen, gliedern sich in Logdaten (Zustände zu einem bestimmten Zeitpunkt) und Antwortdaten (Wert von Variablen zum Ende des Tests). Über das Management-Portal eines Arbeitsbereiches stehen die Rohdaten als Download bereit (CSV mit eingebetteten JSON-Datenstrukturen), über die API können auch reine JSON-Daten abgerufen werden.

## *Review, Demo*

Für Demonstrationszwecke oder zur Prüfung der Testinhalte bzw. des Testablaufes stehen weitere Durchführungsmodi zur Verfügung. Dann werden z. B. keine Antworten in die Datenbank gespeichert, Navigationsbeschränkungen usw. sind aufgehoben oder es steht eine Kommentarfunktion zur Verfügung.

## *System-Check*

Für die Prüfung, ob die technische Ausstattung vor Ort (in der Schule) den Mindestvoraussetzungen der Durchführung entspricht (Netzverbindung, Browser), steht im Testcenter ein sog. System-Check

zur Verfügung. Dieser kann durch entsprechende Parameter auf eine konkrete Testung abgestimmt werden.

## Hinweise zur Integration

- Authentifizierung: Bei jedem Login mit einem Account für den Zugriff auf Arbeitsbereiche wird ein neues AuthToken erzeugt und das vorherige AuthToken dieses Accounts wird ungültig. Das AuthToken wird im http-Header in einem Parameter mit einem eigenen Namen geführt.
- Die XML- Formate für den Testinput sind stabil. Sie werden ggf. für neue Funktionen erweitert, aber die derzeitige Definition kann als erprobt angesehen werden. Einige Bereiche der Definition werden wenig bis kaum genutzt (z. B. mehrstufiges Login), verbleiben aber in der Definition.
- Die API für das Hochladen des Testinputs ist ebenfalls seit längerer Zeit konstant.

## Ausblick

Folgende Änderungen bzw. Erweiterungen sind für 2023 geplant:

- Implementierung von Cache-Strategien zur Verbesserung des Lastverhaltens
- Umstellung des Auth-Verfahrens für Admin-Zugriffe auf JWT (bearer)
- Endpunkte für Health-Check: Für das Servermonitoring werden API-Endpunkte eingeführt, die über den technischen Zustand des Systems berichten (CPU-Auslastung, RAM, Volumes, Datenbank alive? usw.)
- Orchestrierung über Kubernetes ermöglichen für Skalierbarkeit und Ausfallsicherheit

Folgende Änderungen bzw. Erweiterungen sind nach 2023 geplant:

- Implementierung VIDIS/OpenID Connect
- Adaptive Testdurchführung: Die Aufgabenfolge ist nicht mehr statisch, sondern richtet sich nach vorherigen Antworten. Dazu wird eine Schätzung des Leistungsniveaus der Testperson während der Beantwortung vorgenommen.
- Erweiterung der API für den Download von Antwortdaten für bessere Auswertung

## Datenauswertung

### Rohdaten

Die Daten aus dem Testcenter bzw. aus der Einbettung eines Verona-Players bezeichnen wir als Rohdaten. Da die Verarbeitung von Log-Daten im Moment kaum ausformuliert ist, beziehen sich die folgenden Ausführungen auf die Antwortdaten – also den letzten Stand der Beantwortung durch die Testperson beim Beenden bzw. Verlassen des Tests.

- Player: In der Verona-Spezifikation ist die Antwort einer Testperson als Text (string) definiert, d. h. das Datenformat ist nicht durch den Standard definiert. Defacto sind das natürlich serialisierte Datenstrukturen. Die Player des IQB liefern eine einheitliche flache Datenstruktur „IQB-Standard“, die von den anschließenden IQB-Verarbeitungsschritten verstanden wird.
- Das Testcenter fügt zu den Antwortdaten des Players noch die Daten der Testperson (Loginname, Gruppe usw.), der Aufgabe (ID) sowie einen Zeitstempel hinzu. Die Antwortdaten werden an dieser Stelle nicht verändert.

## **Aufbereitung**

Wer sinnvoll mit den Daten arbeiten will, braucht zunächst eine Matrix Person x Variable -> Wert. Die Variablenbezeichnungen aus den Playerdaten sind nicht unbedingt eindeutig (unique) im Test, d. h. die Aufgaben-ID bzw. -Alias muss oft zu der Variablen-ID hinzugefügt werden. Das IQB stellt für sein Format „IQB-Standard“ eine Software bereit, um die Text-Version der Playerdaten in die Auswertungsmatrix zu überführen.

## **Kodierung**

Der Begriff „Kodierung“ steht für die Übersetzung der Antwort in einen numerischen Wert, der die Antwort ab diesem Zeitpunkt symbolisiert („Code“). Auf diese Art wird die Antwort einer „Kategorie“ zugewiesen, wodurch in einem zweiten Schritt die Bewertung der Antwort möglich wird („Score“). Außerdem erfolgt mitunter die Zusammenführung mehrerer Antworten zu einem Auswertungsaspekt (sog. „Item“).

Wir unterscheiden zwischen automatisierter und manueller Kodierung. Viele Antwortformate sind geschlossen, z. B. „Auswahl aus einer Liste“. Kodierung und Bewertung kann dann über einen einfachen Algorithmus erfolgen. Wenn jedoch eine Automatisierung nicht oder nur mit einer gewissen Wahrscheinlichkeit möglich ist, muss die Antwort einer Person „Kodierer:in“ vorgelegt werden (Liste mit Antworten oder Laden der Antwort/Rohdaten in einen Player).

Die Tradition der papierbasierten Testung sieht bei VERA vor, dass die Lehrkräfte die ausgefüllten Testhefte sichten und nach einer Vorschrift („Kodieranweisungen“) kodieren und gleich bewerten. Die Bewertung wird dann in ein Portal eingegeben, das vom Land bzw. der beauftragten Einrichtung bereitgestellt wird. Beim computerbasierten Testen muss ebenfalls ein ähnliches Portal vorgesehen werden, auch wenn die Kodierung zentral und nicht durch die Lehrkräfte erfolgt. Das IQB wird 2023 die Webanwendung „IQB-Kodierbox“ für diesen Zweck entwickeln.

Die Kodieranweisungen liegen zunehmend im digitalen Format vor, um die automatische Kodierung und Bewertung zu ermöglichen. Das Datenformat hierzu befindet sich in Entwicklung, ebenso wie Komponenten zur Unterstützung der automatischen Kodierung.

Die Daten, die zum Ende des Arbeitsschrittes „Kodierung“ vorliegen, nennen wir „Primärdaten“.

## **Auswertung**

Bei Lernstandserhebungen werden anschließend aus den Primärdaten die personenbezogenen Kompetenzniveaus ermittelt. Hierzu sind die Itemkennwerte der verwendeten Aufgaben heranzuziehen, um die Leistungsdaten auf einer einheitlichen Metrik abzubilden. Anschließend gehen Vergleichswerte in die Analyse ein (andere Schulen, andere Bezirke, Landesdurchschnitt usw.), so dass in der Rückmeldung ein Bild geliefert werden kann, das der Unterrichtsentwicklung dient.

## **Ausblick**

Es liegen Erfahrungen aus den papierbasierten Tests vor, aber auch schon aus computerbasierten Erhebungen. Bisher war hier viel Handarbeit nötig, d. h. Daten wurden manuell transformiert, aggregiert und analysiert. Ab 2023 beginnen zwei Projekte, die über den DigitalPakt Schule gefördert werden, die bisherigen Erfahrungen zusammenzutragen, Vorschläge für Datenformate und Auswertungsverfahren zu entwickeln sowie entsprechende Komponenten zu programmieren.

## Empfehlungen

- Synergien suchen: Das beschriebene technische System eignet sich nicht nur für Lernstandserhebungen, sondern man kann damit auch Befragungen unterschiedlicher Art oder Evaluationen durchführen. Auch Prüfungen wären prinzipiell möglich, wenn man weitere Rahmenbedingungen schafft (z. B. eingeschränkter Browser SEB).
- Studio installieren: Die Webanwendung „IQB-Studio“ ist zwar auf die komplexen Arbeitsabläufe des IQB abgestimmt, aber sie lässt sich auch gut im Land für die Nachbearbeitung von VERA-Aufgaben oder für das Anlegen neuer Seiten/Aufgaben verwenden.
- Schätze bergen: Sollten digital umfangreiche Aufgabensammlungen existieren z. B. für das automatisierte Erzeugen von papierbasierten Testinstrumenten, kann durch die Programmierung eines passenden Players schnell der Einsatz in einem computerbasierten Test erreicht werden.
- An Entwicklung beteiligen: Der Open Source-Ansatz lädt dazu ein, die Anwendungen des IQB bzw. die Schnittstellen und Datenformate gemeinsam weiterzuentwickeln.
- Verantwortung übernehmen: Die Lizenz MIT schafft maximale Freiheit der Verwendung, stellt aber auch das IQB frei von der Haftung für die Folgen von Fehlfunktionen. Es erhöht sich damit die Pflicht der Anwendungsseite, Zeit und Ressourcen für Qualitätsprüfungen im Vorfeld einer jeden Lernstandserhebung bereitzustellen.