



Von Excel zu Exzellenz: Pragmatischer Einstieg in Self-Service BI mit iterativer Architektur

In vielen Unternehmen dient Excel als gewachsener Standard für Controlling-Aufgaben. 74% der deutschen Firmen mit einem Umsatz über 100 Mio. Euro nutzen es intensiv für Datenkonsolidierung und -analyse, stoßen jedoch an Skalierbarkeitsgrenzen wie fehlende Automatisierung und Versionskonflikte.

Dieser Artikel zeigt einen innovativen SSBI-Ansatz, der Excel-Nutzer dort abholt, wo sie stehen, und pragmatisch in eine moderne Layer-Architektur überführt. Der Clou: Die Datenmodellierung erfolgt ausschließlich auf dem Kundenhauptsystem mittels SQL-Views. Auf teure Zwischenschichten wird zunächst verzichtet. Ergänzt wird dieses technische Gerüst durch ein klares organisatorisches Framework: Mut zur Unperfektion, eine offene Fehlerkultur und die Bereitschaft, Prototypen konsequent zu verwerfen.

Dieser Beitrag verbindet technische Skalierbarkeit mit kultureller Reife und beantwortet die zentrale Praxisfrage: Wie gelingt eine SSBI-Transformation, die gesund wachsen kann?

Methodischer Ansatz für den Einstieg in moderne BI-Architekturen

Unternehmen stehen beim Aufbau einer BI-Lösung grundsätzlich vor zwei Gegenpolen: dem direkten Einstieg in eine umfassende Enterprise-Architektur oder einem pragmatischen, Use-Case-zentrierten Vorgehen aus der bestehenden Excel-Welt heraus. Beide Wege adressieren dieselbe Zielsetzung: eine skalierbare, wartbare und

vertrauenswürdige Daten- und Analyseplattform. Sie unterscheiden sich jedoch massiv hinsichtlich Investitionshöhe, Komplexität und Lernkurve.

Eine klassische Enterprise Solution setzt in der Regel auf ein vollständiges Architekturkonzept mit Data Warehouse oder Lakehouse, ETL-Strecken, Governance-Modell und einem standardisierten Reporting-Frontend. Für Organisationen, deren Controlling und Reporting bislang fast

ausschließlich in Excel stattfindet, ist dieser Ansatz häufig mit drei Problemen verbunden:

- Erstens stellt er eine hohe finanzielle Einstiegshürde dar, ohne dass der erwartete Mehrwert und der Zeitpunkt des ROI seriös beziffert werden können.
- Zweitens fehlen oftmals grundlegende Erkenntnisse darüber, welche Tabellen, Dateien und Berechnungs-

logiken im bestehenden Excel-Ökosystem tatsächlich geschäftskritisch sind.

- Drittens lässt sich das zukünftige „Kernstück“ einer Schichtenarchitektur – also die fachlich konsolidierte, bereinigte und historisierte Datengrundlage – weder sauber abgrenzen noch zielgerichtet aufbauen.

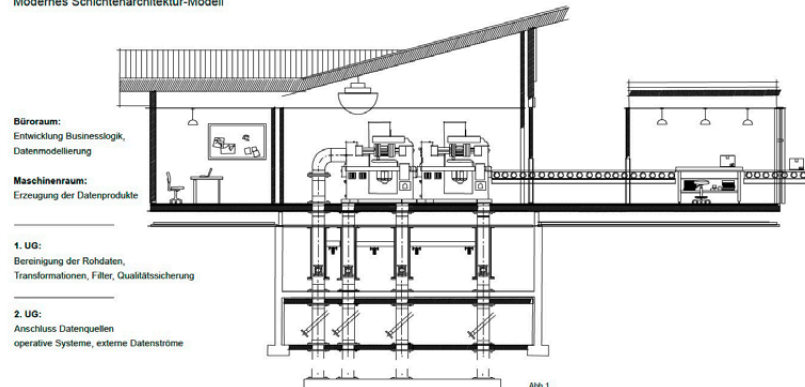
Als Gegenentwurf bietet sich ein Use-Case-fokussierter Self-Service-Ansatz an. In der Praxis führt Recherche zu Self-Service BI in Microsoft-Umgebungen sehr schnell zu Power BI als de-facto-Werkzeug der Wahl. Der übliche Einstieg erfolgt dabei über die Datenmodellierung im Power Query Editor. Dieser ist bewusst niedrigschwellig konzipiert: Datenquellen lassen sich ohne Programmierkenntnisse anbinden, bereinigen und kombinieren. Transformationen lassen sich weitgehend per Oberfläche konfigurieren. Für erste Prototypen und kleinere Fachbereichslösungen ist das äußerst attraktiv.

Genau diese Stärke wird jedoch beim schrittweisen Ausbau zum Problem. Zum einen stößt der Power Query Editor – insbesondere in Import-Szenarien – verhältnismäßig schnell an technische Grenzen, etwa bei Speicherbeschränkungen oder steigender Modellkomplexität. Zum anderen ist die in Power Query implementierte Businesslogik kaum skalierbar: Sie ist in einzelnen Berichten oder Datenmodellen gekapselt, schwer wiederverwendbar und nur mit erheblichem Aufwand test- oder versionsfähig. Hat man eine professionelle Schichtenarchitektur als Zielbild („Nordstern“) im Blick, bedeutet ein reiner Power-Query-getriebener Einstieg häufig, dass wesentliche Teile der Logik später neu interpretiert, konsolidiert und in andere Schichten überführt werden müssen.

Die Logik gehört in die Datenbank

Die logische Konsequenz ist ein Ansatz, der Self-Service-Fähigkeit und

Modernes Schichtenarchitektur-Modell



spätere Enterprise-Tauglichkeit verbindet, statt sie gegeneinander auszuspielen. Der Kern dieses Ansatzes ist eine innovative, aber in sich stringente Herangehensweise: Die fachliche Modellierung erfolgt von Anfang an dort, wo sie langfristig liegen soll – auf Ebene der zentralen Datenquelle, idealerweise eines SQL-Servers. Die Self-Service-Komponente (z. B. Power BI) wird primär zur Ausprägung des semantischen Modells und der Reporting-Oberfläche genutzt, nicht als Ort der Kernlogik. Voraussetzung ist, dass sich der betrachtete Use Case im ersten Schritt weitgehend auf eine Quelle abstützen lässt, typischerweise eine operative oder analytische SQL-Datenbank.

In diesem Modell bildet der SQL-Server den Ort, an dem Struktur, Integrationslogik und Businessregeln definiert und iterativ geschärft werden. Views, Stored Procedures oder andere serverseitige Artefakte repräsentieren nach und nach genau jene Schichten, die später Teil einer voll ausgebauten Architektur sein sollen. Power BI konsumiert diese vorbereiteten Datenmodelle und ergänzt sie um ein schlankes semantisches Modell. Hier wird nur ergänzt, was nicht im Vorkonzept schon gelöst werden kann – wie Zeitintelligenz und weitere Berechnungen in DAX, die Zusammensetzung des Datenmodells, Rollen und Berechtigungskonzepte. Das wird den Fachanwendern in Form von Berichten und Dashboards zur Verfügung

gestellt. Auf diese Weise bleibt die Einstiegshürde gering – es wird weiterhin mit vertrauten Werkzeugen gearbeitet –, gleichzeitig erfolgt der Wissensaufbau dort, wo er die spätere Enterprise-Lösung direkt vorbereitet: in der Struktur der Datenbank und im Verständnis der eigenen Prozesse und Kennzahlen.

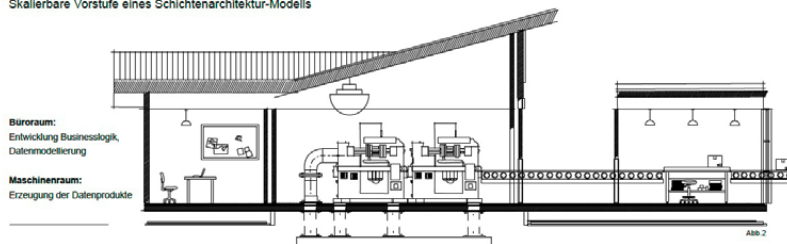
Der Nordstern: Moderne Schichtenarchitektur als Zielbild

Die moderne Schichtenarchitektur lässt sich als Industriegebäude visualisieren:

- **Im Fundament** werden Rohdaten aus verschiedenen Quellen angebunden und technisch aufbereitet – bereinigt, transformiert und mit Qualitätskontrollen versehen.
- **Das Erdgeschoss** enthält die fachliche Schicht: Hier entsteht die Businesslogik mit KPIs und Planungsmodellen, aus denen standardisierte Datenprodukte erzeugt werden.
- **Über ein Fließband** gelangen diese in ein separates Frontend-Gebäude (Power BI), wo sie zu konsumfertigen Reports verpackt und an Endnutzer ausgeliefert werden.

Dieses Bild beschreibt den Nordstern einer modernen Schichtenarchitektur: klar getrennte Verantwortlichkeiten, klare Übergabepunkte und eine nachvollziehbare Wertschöpfungskette von der Datenquelle bis zum Report.

Skalierbare Vorstufe eines Schichtenarchitektur-Modells



Der innovative Ansatz: Erdgeschoss zuerst, aber richtig

Der in diesem Artikel vorgestellte Weg setzt genau hier an, jedoch mit einem bewusst reduzierten Startumfang. Anstatt die komplette „Untergeschossebene“ mit allen Datenquellen und Transformationen auf einmal zu bauen, wird zunächst nur das Erdgeschoss des Modells materialisiert. Die einzige, klar definierte Datenquelle ist eine SQL-Datenbank. Diese Quelle übernimmt in der Anfangsphase die Rolle sowohl der Rohdatenablage als auch der Transformations- und Logikschicht (Abb.2).

Im übertragenen Bild bedeutet das: Das Planungsbüro entwickelt unmittelbar auf der angebundenen SQL-Datenbank die Businesslogik – über Views, Joins, Aggregationen und berechnete Felder. Die Maschinen im selben Geschoss erzeugen daraus die ersten Datenprodukte, die direkt an ein semantisches Modell in Power BI übergeben werden. Der Power Query Editor spielt in diesem Vorgehen keine Rolle; stattdessen werden Transformation und Modellierung bewusst dort verankert, wo sie später Teil der professionellen Schichtenarchitektur sein sollen.

Dieser Ansatz erlaubt es, mit einem konkreten, fachlich relevanten Use Case pragmatisch zu starten, ohne in den typischen Lock-in eines rein toolzentrierten Self-Service-Starts zu geraten. Gleichzeitig entstehen entlang des Weges Artefakte, Strukturen und ein tiefes Verständnis der eigenen Datenlandschaft, die sich nahtlos in eine spätere Enterprise-Architektur überführen lassen. Damit wird die Lücke zwischen Excel-getriebenem Controlling und einer modernen,

mehrschichtigen BI-Plattform nicht durch einen Big Bang, sondern durch einen kontrollierten, methodisch geführten Lern- und Reifeprozess geschlossen.

Iterativer Entwicklungsprozess: Die fünf Säulen

Der beschriebene methodische Ansatz lebt vom konkreten Umsetzungsprozess, der sich als Säulendiagramm mit fünf Phasen visualisieren lässt (Abb.3). Jede Säule repräsentiert einen Entwicklungsschritt, wobei die Höhe die erreichte Reife und Qualität widerspiegelt. Säule 1 bildet die niedrige Basis des Wissensaufbaus, Säule 5 den ambitionierten Nordstern. In Säule 2 geht es um einen pragmatischen Start mit einem MVP. Säulen 3 und 4 wiederholen sich iterativ, bis das Ziel erreicht ist – ein Prinzip, das an Throwaway-Prototyping und agile BI-Entwicklung angelehnt ist.

Säule 1: Wissensaufbau als Fundament

Der Einstieg beginnt zwangsläufig mit geringer Höhe, da in Excel-do-

minierten Umgebungen meist weder Datenstrukturen noch fachliche Modelle systematisch dokumentiert sind. Hier sind gezielte Schulungen zu Power BI und SQL-Server essenziell, um die Grundlage für den pragmatischen Start zu legen.

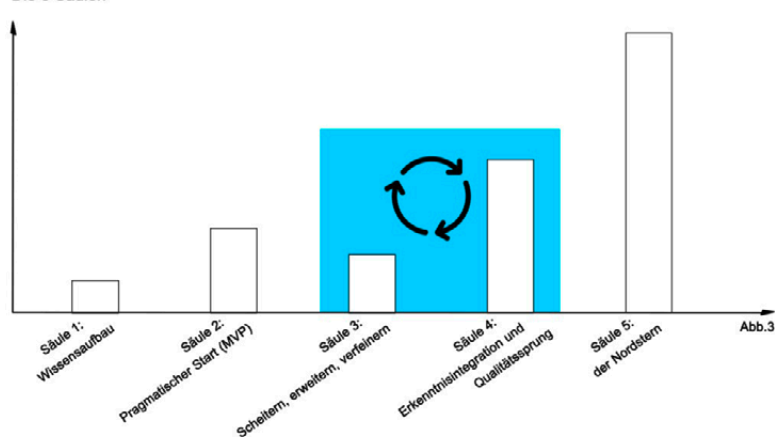
Säule 2: Pragmatischer Start mit Mut zur Unperfektion

Mit Basiswissen folgt der erste Prototyp: Use-Case-relevante Daten werden aus der SQL-Quelle erkundet, kontextualisiert und modellierend in Views oder Stored Procedures übersetzt. Hier steht der „Mut zur Unperfektion“ im Vordergrund – ein lauffähiges Modell entsteht schnell, um erste Erkenntnisse zu gewinnen. Dimensionen und Fakten werden definiert, erste Joins getestet, ohne Perfektion anzustreben.

Säule 3: Scheitern, Erweitern, Verfeinern

Keine Entwicklung entgeht dieser Phase. Erste Modelle scheitern an unklaren Anforderungen, fehlender Datenqualität oder falschen Annahmen. Eine offene Fehlerkultur ist entscheidend: Fehler werden nicht sanktioniert, sondern als Hebel zur Verbesserung genutzt. Hier entfaltet sich der volle Wert des Ansatzes – der Prototyp wird analysiert, Lücken identifiziert und refraktioniert. Dies entspricht dem „Minimum Viable Product“ (MVP).

Die 5 Säulen



Säule 4: Erkenntnisintegration und Qualitätssprung

Die Iterationen-Säulen 3/4 führen zu nachhaltigem Fortschritt. Erkenntnisse fließen ein, Modelle werden verschlankt, kommentiert und erweitert. Säule 4 übertrifft Säule 2 in Qualität, da sie auf Erfahrung basiert. Die Anzahl und Dauer dieser Schleifen hängen von mehreren Einflussfaktoren ab, die den Prozess beschleunigen oder verzögern.

Ein zentraler Hebel ist der Umgang mit Prototypen: Statt endlos zu pflegen, empfiehlt die Methodik „Wegschmeißen und neu machen“. Dies wirkt zunächst kontraintuitiv, liefert jedoch zwei Effekte: Erstens entsteht eine schlanke, reflektierte Neustrukturierung ohne technische Schulden. Zweitens ermöglicht der Abgleich mit dem Vorgänger kritische Validierung. Abweichungen werden leicht festgestellt und gefixt, da man nicht mehr nur mit der Excel Welt abgleicht, sondern schon mit einer fortgeschrittenen Aufbereitung.

Säule 5: Nordstern – Die moderne Layer-Architektur

Alle Iterationen orientieren sich am Zielbild der modernen Layer-Architektur. Säule 5 repräsentiert die vollständige Schichtenarchitektur mit klarer Trennung von Rohdaten, Transformation, Businesslogik und konsumfertigen Reports. Sie wird nie als Big Bang erreicht, sondern schrittweise materialisiert.

Einflussfaktoren auf Iterationserfolg

Die Effizienz des Prozesses wird durch vier Schlüsselvariablen gesteuert:

1. Schulungsqualität

Die Qualität der initialen Schulung (Säule 1) bestimmt maßgeblich, wie steil die Lernkurve im weiteren Prozess verläuft. Reine Tool-Trainings konzentrieren sich fast ausschließlich auf Klickpfade in Power BI. Sie fördern das Missverständnis, Power Query sei lediglich das modernere Excel und da-

mit der zentrale Ort aller Transformationen. Dadurch verlagern sich fachliche Logik und Integrationsaufgaben in ein Werkzeug, das nie für dauerhafte Kernlogik konzipiert war, was den späteren Migrations- und Iterationsaufwand deutlich erhöht. Wesentlich wirksamer sind Schulungsformate, die Architekturdenken, Datenmodellierung (Fakten/Dimensionen) und das übergeordnete Zielbild in den Mittelpunkt stellen. Das Tool ist nur ein Mittel zum Zweck. Wer von Beginn an versteht, wie ein semantisches Modell auf einer sauberen Datenbasis aufsetzt, kann die ersten Schritte bewusst planen und spart sich viele nachgelagerte Schleifen.

2. Teamdynamik und externe Expertise

Wie effizient Säule 3 und 4 durchlaufen, hängt stark von der Zusammensetzung des Teams und der verfügbaren Unterstützung ab. Wenn Unerfahrene mit dem Aufbau des SSBI experimentieren, werden zwar viele Erfahrungen gesammelt, doch jede Erkenntnis muss mühsam erarbeitet werden, was die Dauer und Anzahl der Iterationen erhöhen. Kommt ein Experte dazu, lassen sich typische Fallstricke früh erkennen und vermeiden, ohne sie alle selbst durchexerzieren zu müssen.

Entscheidend ist auch, wie das Team miteinander arbeitet: Offene Kommunikation, regelmäßige Feedback-Runden und eine bewusst begrenzte Zahl aktiver Mitwirkender sorgen dafür, dass der Fokus gewahrt bleibt. Beteiligte sollten bereit sein zu lernen und in Modellen abstrakt zu denken. In größeren Gruppen



Ines Labusch, Consultant

Data Management, Analytics, Visualization, QUNIS GmbH

braucht es eine klare Moderation, die Diskussionen kanalisiert und verhindert, dass sich das Projekt in Detaildebatten verliert.

3. Governance als Change Management

Data Governance wird oft als bürokratische Hürde wahrgenommen – zu Recht, wenn sie rein administrativ gedacht ist. Der Paradigmenwechsel liegt darin, Governance-Verantwortliche als Change Manager zu sehen: Sie etablieren Prozesse, besetzen Rollen, überwachen Reviews und sichern Nachhaltigkeit. Dies begleitet den Aufbau und erzeugt als Nebeneffekt kulturellen Wandel: Abteilungsübergreifende Zusammenarbeit, gemeinsame Daten-Sprache, Motivation zu Datenqualität. Der „Hunger nach mehr“ entsteht dabei organisch, nicht erzwungen.

4. Lösch- und Review-Kultur

Ein oft unterschätzter Faktor für die Qualität einer BI-Lösung ist der bewusste Umgang mit bestehenden Artefakten. Ohne etablierte Review-Routinen häufen sich Berichte, Da-

tenmodelle und Kennzahlen an, die irgendwann niemand mehr aktiv nutzt, die aber weiter gepflegt und mitgeschleppt werden. Eine gelebte Lösch- und Review-Kultur setzt genau hier an: In regelmäßigen Abständen wird kritisch hinterfragt, welche Reports, Seiten oder Modelle tatsächlich noch benötigt werden und welchen Beitrag sie zum Zielbild leisten. Hierzu gibt es auch Unterstützung z.B. durch Log Analytics oder Eventhouse Monitoring. Das gezielte Entfernen veralteter oder redundanter Artefakte ist dabei kein Verlust, sondern ein Qualitätsfilter – fällt ihr Wegfall niemandem negativ auf, schaffen sie eher Ballast als Mehrwert. So bleibt die Lösung schlank, nachvollziehbar und auf den Nordstern ausgerichtet, anstatt in einem schwer wartbaren Stückwerk zu enden.

Fazit des Prozesses

Dieser iterative Ansatz transformiert Excel-Chaos nicht nur technisch, sondern kulturell. Durch bewusste Prototypen, radikale Neuentwicklungen und Governance als Enabler entsteht eine skalierbare Lösung mit tragfähiger Organisationsreife. Die Säulen sind kein lineares Modell, sondern eine sich wiederholende Lernspirale, deren Effizienz von der Initialausrichtung und Begleitung abhängt.

Zusammenfassung und Ausblick

Der hier vorgestellte Ansatz transformiert den Übergang von Excel zu einer modernen Architektur in einen kontrollierten, lernenden Prozess. Er verbindet einen pragmatischen Einstieg mit disziplinierter Methodik. Praktisch bedeutet das: Unternehmen erreichen schneller einen ROI, da Use

Cases lauffähig werden, während sie die Enterprise-Architektur pflastern. Der kulturelle Wandel – Motivation zu Datenqualität, „Hunger nach mehr“ – entsteht organisch durch Mitwirkung, nicht durch Dekret.

Nächste Schritte: Identifizieren Sie einen Pilot-Use Case mit einheitlicher SQL-Quelle. Starten Sie mit architekturorientierter Schulung (Säule 1), bauen Sie den ersten Prototyp (Säule 2) und etablieren Sie ein Governance-Team als Change-Manager. Iterieren Sie konsequent, werfen Sie weg, verfeinern Sie – bis der Nordstern materiell wird.

Dieser Weg professionalisiert Self-Service BI nachhaltig: von Chaos zu Vertrauen, von Excel zu Excellence.

Ines Labusch

Literaturverzeichnis

Primärquellen (theoretische Abhandlungen zu SSBI, Governance & Architektur)

1. Lehn, K. (202x): Effiziente Anwendungen von Self-Service Business Intelligence. Bachelorarbeit, Leibniz Universität Hannover. Verfügbar unter: https://www.iwi.uni-hannover.de/fileadmin/iwi/Abschlussarbeiten/sK_BA_Lehn.pdf
2. Fischer, F.B.; Burger, A.A.; Gehling, B. (2023): Aktuelle Herausforderungen bei der Implementierung von Self-Service Business Intelligence. HMD Praxis der Wirtschaftsinformatik. DOI: 10.1365/s40702-023-00962-4. Verfügbar unter: <https://edoc.ku.de/id/eprint/320191/s40702-023-00962-4.pdf>
3. Pi-Architektur-Team (2019): Pi-Architektur – Agiles Datenmanagement in Big-Data-Umgebungen. SIGS. Verfügbar unter: <https://www.sigs.de/artikel/pi-architektur/> [sigs]
4. MIP GmbH (2026): Self-Service BI: Theorie und Realität im Check. Verfügbar unter: <https://www.mip.de/wissen-und-praxis/self-service-bi-theorie-und-realitaet-im-check>
5. DataMart (2026): Der erfolgreiche Weg zu einer modernen Datenarchitektur. Verfügbar unter: <https://www.datamart.de/der-erfolgreiche-weg-zu-einer-modernen-datenarchitektur/>

Sekundärquellen (Methodik, Governance, Kulturwandel)

6. Alpar, P.; Schulz, M. (2016): Self-Service Business Intelligence. Business & Information Systems Engineering, 58(2), S. 151–155. Zitiert in [sigs]
7. Lennerholt, C. et al. (2018): Literaturreview zu SSBI. Zitiert in <https://www.econstor.eu/bitstream/10419/311001/1/s40702-023-00962-4.pdf>
8. Throwaway Prototyping (Methodik): Claritee Blog (2024): Throwaway Prototyping: When and How to Use It. Verfügbar unter: <https://claritee.io/blog/throwaway-prototyping-when-and-how-to-use-it/>

9. SR Analytics (2025): Self-Service BI: Build a Data-Driven Culture Fast. Verfügbar unter: <https://sranalytics.io/blog/self-service-bi-guide/>
10. GP Strategies (2025): 5 Strategies to Build a Data-Driven Culture Through Governance. Verfügbar unter: <https://www.gpstrategies.com/blog/5-strategies-to-build-a-data-driven-culture-through-governance/>
11. Detecon (2025): Data Change Management: Two Elements of the Data-Centric Organizational Culture. Verfügbar unter: <https://www.detecon.com/en/insights/article/data-change-management-two-elements-of-the-data-centric-organizational-culture>

Praktische Ressourcen (Implementierung & Best Practices)

12. Prometheus AI (2026): Self-Service Analytics Implementation: Strategy, Governance & Best Practices. Verfügbar unter: <https://promethium.ai/guides/self-service-analytics-strategy-implementation/>
13. Metis BI (2025): Power BI Governance: Balancing Control and Self-Service for Adoption. Verfügbar unter: <https://www.metisbi.co.uk/blog/power-bi-governance-balancing-control-and-self-service-for-adoption>
14. OMR Reviews (2023): BI-Implementierung: Fünf Schritte für eine erfolgreiche Einführung. Verfügbar unter: <https://omr.com/de/reviews/contenthub/bi-implementation>
15. Kaizen (2025): Business-Intelligence-Strategie. Verfügbar unter: <https://kaizen.com/de/einblicke/business-intelligence-strategie-daten/>

Hinweis: Alle Inline-Zitationen im Text (z. B.) verweisen auf diese Liste.

➤ Zum Selbst-Check Self-Service Analytics
<https://lp.qunis.de/lp-self-service-analytics#formular>