CORE®

Anti Aging for IT

Flexibilität erhalten – Erneuerung managen

Tatsiana Bychkouskaya
Christian Böhning
Dr. Carsten Wedekind
Karsten Trostmann

April 2020
Blogpost

Key Facts

- Die immer schnellere technische Entwicklung der letzten Jahre verursacht einen beschleunigten Alterungsprozess von IT-Systemen.
- Gleichzeitig sorgt die Umsetzung neuer Anforderungen für eine immer stärkere Abweichung der IT-Systeme von ihrer ursprünglichen Konzeption. Ohne entsprechende Gegenmaßnahmen verlieren Systeme ihre konzeptionelle Integrität.
- Integritätsverlust führt zu mehr Komplexität der Systeme, verschlechtert die Änderbarkeit und führt zu steigenden Wartungskosten.
- Der Ansatz einer evolutionären Architektur versucht den Alterungsprozess der IT-Systeme zu verlangsamen und seine Auswirkungen zu begrenzen.
- Wichtiger Bestandteil der evolutionären Architektur ist das Erfassen und Managen der technischen Schulden von IT-Systemen.

Bericht

Unser Leistungsportfolio als strategische IT-Unternehmensberatung beinhaltet auch die Analyse und Bewertung des technischen Zustands von IT-Systemen. Diese Architekturreviews werden meist ausgelöst durch schlechte Anpassbarkeit der IT-Systeme an neue Anforderungen oder mangelhafte Performanz und Qualität in der Erfüllung umgesetzter Anforderungen und die damit einhergehende Unzufriedenheit der Systemnutzer. Im Ergebnis eines Reviews müssen wir oftmals die schlechte Botschaft überbringen, dass die untersuchten IT-Systeme entweder veraltet sind oder sich in einem technisch sehr schlechten Zustand befinden. Die dauerhafte Lösung wäre eine umfassende Restrukturierung oder der vollständige Ersatz der Systeme. Beide Optionen würden zeit- und ressourcenintensive Projekte erfordern, die vorhandene Kapazitäten vollständig binden und die fachliche Weiterentwicklung der Systeme erschweren.

Die IT-Verantwortlichen der betroffenen Systeme stellen sich zurecht die Frage, wie diese Situationen eintreten konnte. In einem konkreten Fall konnten sich die Verantwortlichen nicht erklären, warum ihr mit nicht mal 20 Jahren doch relativ junges, von Grund auf sauber durchmodelliertes und überwiegend in Java implementiertes IT-System ständig unter Qualitätsproblemen litt sowie Änderungen komplex und damit langwierig sind. Die wesentlichen Gründe hierfür waren:

- Seit Inbetriebnahme des Systems im Jahr 2000 wurde bei Erweiterungen und Anpassungen immer stärker vom ursprünglichen Konzept und Entwurf des IT-Systems abgewichen. Aus einem ursprünglichen sauberen und stringenten Design war ein Mix aus verschiedenen Konzepten und Frameworks geworden.
- Das System wurde in seiner bisherigen Lebenszeit fachlich stark ausgebaut. Das beinhaltete auch Funktionen, für die es ursprünglich nicht konzipiert wurde und so zu immer mehr Workarounds, zusätzlichen Schnittstellen in andere IT-Systemen - ergo mehr Abhängigkeiten - führten.

Dieses Muster sind in einer Vielzahl der eingangs erwähnten Architekturreviews zu beobachten. In der Summe lässt sich erkennen, dass IT-Systeme ein Phänomen aufweisen, welches dem menschlichen Alterungsprozess nicht unähnlich ist. Beide werden mit fortschreitendem Lebensalter schwerfälliger und verlieren die Fähigkeit sich schnell an geänderte äußere Gegebenheiten anzupassen. Ihre Leistungsfähigkeit und Zuverlässigkeit nimmt bei gleichzeitigem Anstieg der Aufwände für Pflege und Erhalt ab.

Der Alterungsprozess von IT-Systemen ist dabei kein neues Phänomen. Seine Bedeutung hat aber in den letzten Jahren noch weiter zugenommen. Technologische Innovationszyklen werden immer kürzer und die IT-Systeme spielen mit fortschreiten der Digitalisierung von Prozessen in den Unternehmen eine wesentlich stärkere Rolle in der Wertschöpfung. Um sowohl Auswirkungen als auch Tempo des Alterungsprozesses von IT-Systemen zu begrenzen, müssen dessen Ursachen verstanden und rechtzeitig geeignete Gegenmaßnahmen eingeleitet werden.

Warum veralten IT-Systeme?

Schon in den 90iger Jahren des letzten Jahrhunderts hat sich die Computerwissenschaft mit dem Prozess der Alterung von IT-Systemen und Software befasst. Kein geringerer als David L. Parnas, dem wir unter anderem das für jedes nichttriviale System fundamental wichtige

Kapselungs- und Geheimnisprinzip verdanken, hat in seinem Essay „Software Aging“ die zwei wesentlichen Ursachen dargelegt.

Er führt als erste Ursache an, dass Systeme nach Entwurfsparadigmen entworfen werden, die an den technischen, wirtschaftlichen und organisatorischen Rahmenbedingungen ihrer Zeitepoche ausgerichtet sind. Diese Rahmenbedingungen unterliegen jedoch einem permanenten Wandel, welcher wiederum maßgeblich durch die Technologieentwicklung bestimmt wird. Ausgehend von den einzelnen technischen Innovationen an sich, sind auch daran angepasste Methoden und Vorgehensweisen einem Wandel unterworfen.

In der Betrachtung eines aus vorherrschenden Rahmenparametern abgeleiteten Entwurfsparadigmen von IT-Systemen lässt sich dies anschaulich nachvollziehen. In den 80er und 90er Jahren des letzten Jahrhunderts waren Rechenleistung und Speicherplatz teuer. Die daraus abgeleiteten Entwurfsparadigmen waren das Konzept des Zentralrechners, die maximale Wiederverwendung von Code und Sparsamkeit als oberstes Prinzip der Datenmodellierung. Unter den heutigen Rahmenbedingungen sind diesen Paradigmen höchst problematische Designs entsprungen, unter deren späten Auswirkungen viele Unternehmen leiden. Was damals als Lösungsmuster gut und richtig war, ist unter den heutigen Anforderungen an Flexibilität ein Anti-Pattern. Aktuelle Methoden und Vorgehensweisen agiler Softwareentwicklung sind nicht ohne wesentliche Einschränkungen auf den alten Technologieplattformen und Systemstrukturen der 90er Jahre abbildbar. Die Möglichkeiten zur Effizienz- und Qualitätsverbesserung, welche durch die IT Entwicklung der letzten 20 Jahre geschaffenen wurden, können so nicht voll genutzt werden. In Kombination mit dem gestiegenen Beitrag der IT-Systeme an der Wertschöpfung in den Unternehmen führt dies in letzter Konsequenz zu Wettbewerbsnachteilen.

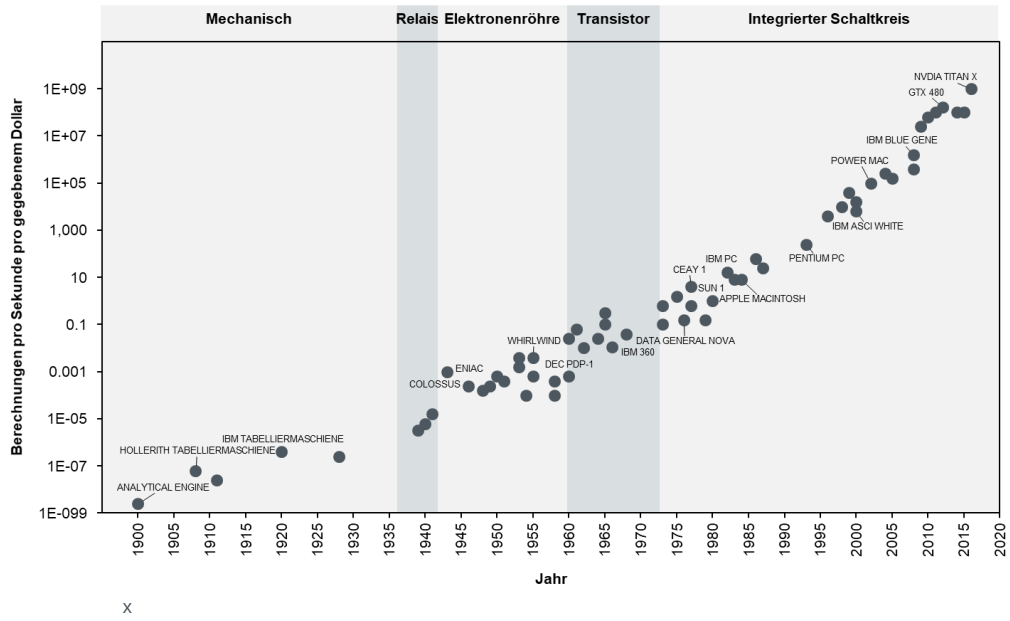

Abbildung 1: Moor‘s Law 1900 bis 2020

Betrachten wir die Entwicklung technischer Rahmenbedingungen am Beispiel der historischen Entwicklung von Rechenleistung, dann sehen wir, dass die Technologieentwicklung ein evolutionäres Verhalten aufweist. Wie bei allen Evolutionsprozessen verläuft diese Entwicklung nicht linear, sondern exponentiell. In der aktuellen Lage des Frühjahrs 2020 ist uns allen sehr

gegenwärtig, was exponentielles Wachstum bedeutet. Die technischen Innovationszyklen werden immer schneller kürzer und in gleichem Maße ändern sich auch Rahmenbedingungen und Entwurfsparadigmen. Die IT-Systeme veralten durch sich stetig beschleunigende Technologieentwicklung also immer schneller.

Die zweite angeführte Ursache für das Veralten von IT-Systeme liegt in der Änderung und Anpassung von IT-Systemen selbst. Für den überwiegenden Teil der in den Unternehmen eingesetzten IT-Systeme wurden und werden umfangreiche Konzepte erstellt, die einer bestimmten fachlichen Zielsetzung und technischen Paradigmen folgten. Zum Zeitpunkt ihrer Inbetriebsetzung entsprechen die Systeme auch dieser Konzeption. Sie weisen also eine hohe konzeptionelle Integrität auf . Der einsetzende Änderungsprozess nimmt aber dann meist einen natürlichen, mehr oder weniger starken entropischen Verlauf: Das System weicht mit zunehmender Zahl an Änderungen und implementierten Funktionen immer mehr vom ursprünglichen Konzept ab. Mit zunehmendem Altem des Systems beschleunigt sich die konzeptionelle Erosion und führt zu einer immer schlechteren Anpassbarkeit und minderer Systemqualität.

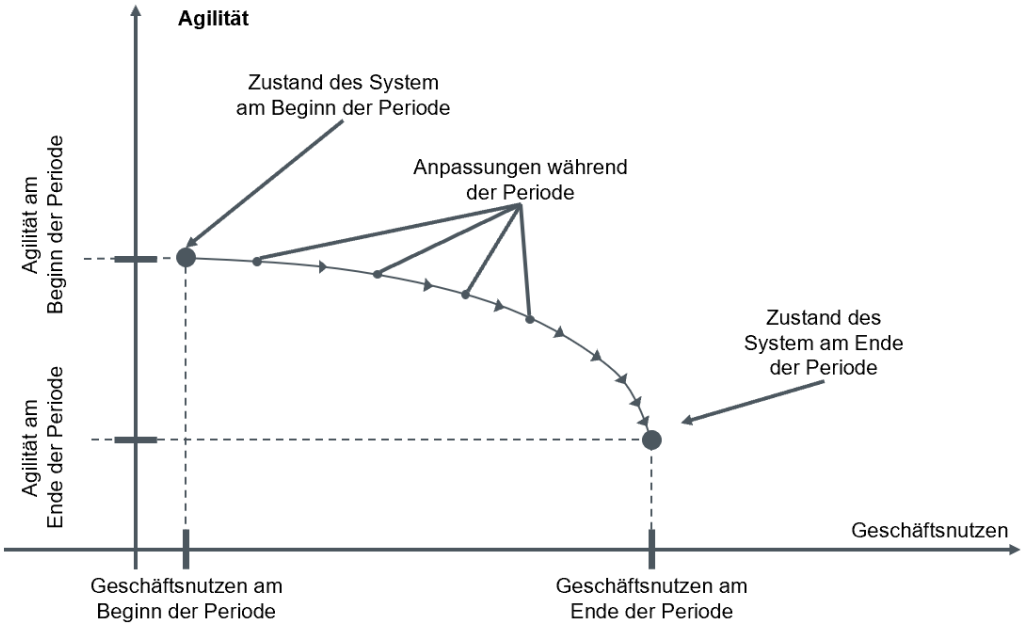

Abbildung 2: Agilität des IT-System und Geschäftsnutzen

Wir wollen einige Hintergründe dieser Errosion aus unserer Sicht darlegen. Der Verlust konzeptioneller Integrität wird in vielen Fällen durch ein ausschließlich auf funktionale Weiterentwicklungen ausgerichtetes Änderungsmanagement verursacht. Änderungen am IT-System sollen opportunistisch mit minimalem Kosten- und Zeitaufwand umgesetzt werden. Die konzeptionelle Integrität des Systems stellt für den Nutzer keinen Wert dar, wird nicht als Zieldimension wahrgenommen und daher nicht gemanagt. Größere Änderungen, vor Allem in monolithischen Systemen mit großen Funktionsbereichen und wechselnden Dienstleistern, führen dann zu regelrechten Erosionsschüben.

IT-Systeme werden meist initial in größeren Projekten, stellenweise durch externe Dienstleister erstellt. Mit der Inbetriebsetzung wird die Software des IT-Systems für gewöhnlich von einem

Projekt- an ein Wartungsteam übergeben. Die Mitglieder des Wartungsteams werden meist erst in späten Projektphasen involviert oder sind nicht Teil des Projektteams. Sie müssen sich dann auf Grundlage einer oft unzureichenden und häufig von fachlichen und architekturellen Konzepten geprägten Dokumentation einarbeiten und mit einer enormen Codebasis vertraut machen. Es hat sich gezeigt, dass sich selbst bei sorgfältigster Dokumentation Transitionsverluste nicht vermeiden lassen und eine „emotionale Verbundenheit“ mit dem Code (noch) fehlt. Änderungen erfolgen pragmatisch auf schnellstem und direktestem Weg, in Unkenntnis oder Ignoranz des ursprünglichen Konzepts.

Ein weiterer und sehr entscheidender Punkt ist, dass mit der abnehmenden konzeptionellen Integrität ein Wertverlust des IT-Systems einhergeht, der nirgendwo erfasst wird, aber wie eine unsichtbare und stetig wachsende Hypothek auf dem System lastet. Nur in den wenigsten Fällen wird dieser Wertverlust registriert und ist den Unternehmen bekannt. Das Problem ist also nicht oder in seinem Ausmaß nur unvollständig bekannt.

Anti-Aging für IT-Systeme

Die Grundlage jeder Form von Steuerung basiert auf konkreten Zahlen und Fakten. Nur so lassen sich entsprechende Handlungsschwerpunkte identifizieren, Maßnahmen planen und diese umsetzen. In der Softwaretechnologie hat sich der Begriff der technischen Schuld als Maß für die Degeneration eines Softwaresystems etabliert.

Die technische Schuld umfasst hierbei die Summe aller technischen Defizite, die sich in dem IT-System über die Zeit eingeschlichen haben. Es handelt sich dabei nicht um konkrete fachliche Fehler in der Software, sondern um verschiedene Aspekte unzureichender Qualität, die sich in folgende Kategorien klassifizieren lassen:

- Design- oder Architekturschwächen (Structure Smells)
- Mängel in der Implementierung (Code Smells)
- Unzureichende Testabdeckungen oder zu geringe Automatisierung (Testschulden)
- Unzureichende Dokumentation

Die technische Schuld drückt dabei den Aufwand der Maßnahmen aus, die notwendig sind, um das IT-System wieder auf das erwünschte Qualitätslevel zu heben. Implementierungs- und Testschulden können recht einfach toolgestützt erfasst werden. Für die Dokumentation ist die Einführung eines formalen und Track Recordings für Architekturentscheidungen (Architectural Decsion Records) hilfreich. Entscheidungen und Konzepte können so später nachvollzogen werden. Sie dienen auch gleichzeitig der Dokumentation des Abbaus struktureller Defizite.

Die Methode des evolutionären Architekturansatzes ist hierbei geeignet zur Erfassung struktureller Schwächen. Kern dieser Methoden sind sogenannte Fitness-Funktionen, die als Architekturszenarien formuliert werden und in regelmäßigen Reviews (alle 3-6 Monate) die strukturelle Integrität der IT-Systeme validieren.

Sind technische Schulden bekannt und erfasst, können entsprechende Maßnahmen getroffen werden. Die Idee eines evolutionären Architekturmanagements verfolgt dabei den Ansatz, technische Schulden innerhalb eines Korridors zuzulassen. Die Interpretation von technischen Schulden als „falsche“ Architekturentscheidungen, die korrigiert werden müssen, ist nicht zielführend. Die technische Schuld lässt sich am besten mit einem Kredit vergleichen. Es kann

und ist durchaus sinnvoll, technische Schulden gezielt zur Ausnutzung von Wettbewerbsvorteilen oder Chancen am Markt aufzunehmen. Sie müssen jedoch erfasst und ihre Auswirkungen begrenzt werden. Nur so kann auch langfristig die Zukunftsfähigkeit der IT-Systeme sichergestellt werden.

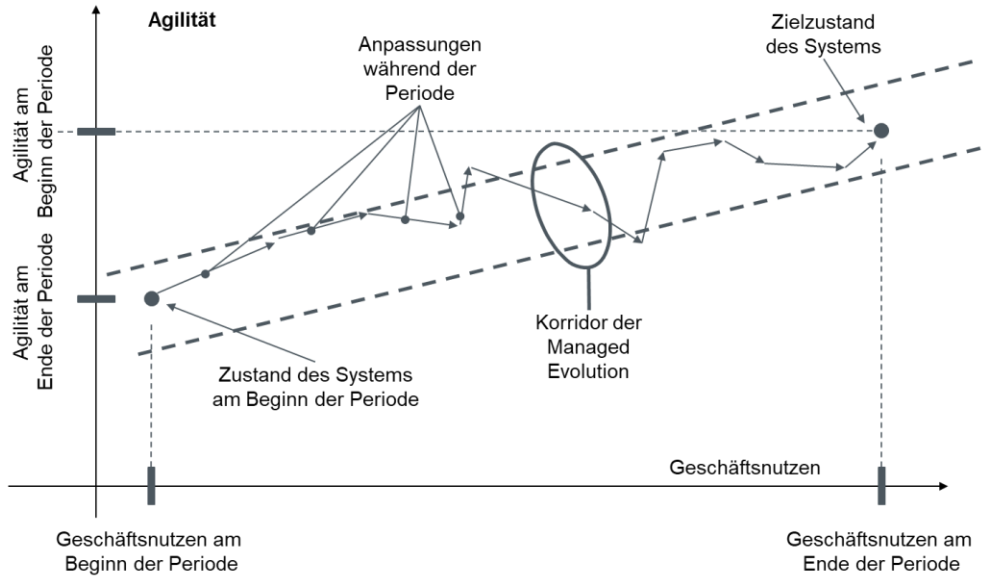

Abbildung 3: Evolutionstrajektorie gemanagter IT-Systeme

Zwei sehr gegensätzliche Prinzipien können angewendet werden, um den Verlust der Anpassungsfähigkeit infolge immer kürzerer Technologiezyklen zu begegnen:

- Wegwerfarchitekturen: schon beim Bau des Systems wird ein konkretes End-of-Live (Verfallsdatum) für das System festgelegt und die Neuimplementierung als technische Schuld mit im Budget eingepreist. Dieser Ansatz bietet sich vor allem für IT-Systeme und Komponenten an, die sehr starken Änderungszyklen unterliegen. Hierzu zählen z.B. Kundenfrontends und APIs.
- Design-to-Change: die Architektur des Systems wird auf maximale Änderbarkeit ausgelegt. Das Prinzip einer hexagonalen Architektur wird dabei mit einer ausschließlich an der fachlichen Funktion orientierten Modularisierung mittels Domain Driven Design kombiniert. Die starke fachliche Modularisierung und technische Entkoppelung macht diese Ansätze aufwendiger in der Implementierung, aber umso robuster gegen Änderungen.

Fazit

Die aktive Gestaltung des Wandels ist auch in Bezug auf die IT-Systeme ein inhärentes Gebot. Dieser Wandel tritt an die IT-Systeme im Besonderen von zwei Seiten heran. Die Mächtigkeit des technischen Fortschritts als Treiber des Wandels lässt sich an der exponentiellen Zunahme der möglichen Rechenoperationen - wie in Moore's Law postuliert - auf einen Blick erkennen. Ein weiterer Treiber des Wandels tritt Außenstehenden selten oder gar nicht in Erscheinung. Das Angebot an Programmiersprachen und Frameworks vergrößert sich ständig und bedingt somit auch eine Erweiterung der zur Verfügung stehenden Architekturen, welche wiederum ihren eigenen Trends und zusätzlichen konzeptionellen Entwicklungen unterliegen. Der technische Wandel bietet hier vor allem eines: Chancen. Der zweite Treiber des Wandels sind die sich stetig

ändernden oder zunehmenden Geschäftsanforderungen. IT-Systeme sind die Möglichmacher des Kerngeschäfts, geänderte Geschäftsanforderungen umzusetzen ist daher dessen zentrale Funktion und im Kontext unbedingt als Chance und weniger isoliert als Störenfried anzusehen.

Die Fokussierung des Wandels auf geschäftsgetriebene Anpassungen ohne begleitende technologische Änderungen brechen die initial bestehende konzeptionelle Integrität des IT-Systems. Ausbleibende Bemühungen diese wieder herzustellen führen zu einer weiteren Degeneration der IT-Systeme, welche sich vor allem in der Agilität weitere Anforderungen zu erfüllen niederschlägt. Der sich in diesem Kontext etablierte Begriff der technischen Schulden impliziert auch, sofern nicht wieder in die konzeptionelle Integrität investiert wird, das unvermeidbare: der Bankrott, das zerschlagen des Tisches und damit Vernichtung der Geschäftsgrundlage.

Die evolutionäre Architekturarbeit kann als das technische Pendant zur Finanzplanung angesehen werden. In diesem Konzept kann die Kompromittierung der konzeptionellen Integrität zur Erfüllung von Geschäftsanforderungen in den höheren Schichten der Systemarchitektur über eine Zeit in Kauf genommen werden, nachgelagert sollte diese durch Anpassungen in den darunterliegenden Schichten wiederhergestellt werden.

Quellen

David L. Parnas. (Mai 1994). Software aging. *Proceedings of the 16th international conference on Software engineering* (S. 279-287). Sorrento, Italien: IEEE Computer Society PressWashingtonDCUnited States.

Murer, S. a. (2010). *Managed Evolution: A Strategy for Very Large Information Systems.* Springer.

Tatsiana Bychkouskaya ist Transformation Associate bei CORE. Ihre Schwerpunkte liegen bei der Strategieberatung für Tech Startups, agilem Projektmanagement, der Konzeption und Implementierung komplexer Projekte im Bereich landesweite IT-Infrastruktur. Tatsiana's Erfahrungen beziehen sich unter anderem auf die Implementierung eines landesweiten Projektes und der Strategieberatung für Start-Ups in der Tech-Industrie.

Mail: tatsiana.bychkouskaya@core.se

Christian Böhning ist Managing Director bei CORE. Er besitzt langjährige Erfahrung in der Durchführung von Technologie-getriebenen Transformatio nen in der Finanzindustrie. Schwerpunkte seiner Arbeit sind Programme zur IT-Architekturmodernisierung, Durchführung von Compliance-Initiativen und die Neuausrichtung von IT-Organisationen.

Mail:christian.boehning@core.se

Als Expert Director bei CORE konzentriert sich **Dr. Carsten Wedekind** auf die strategische Ausrichtung von IT-Architekturen. Er fungiert auch als Co-Vorsitzender der Lending Working Group im Banking Industry Architecture Network (BIAN). Der promovierte Physiker verfügt über langjährige Erfahrung in der Umsetzung von IT-Projekten mit agilen und klassischen Methoden.

Mail: carsten.wedekind@core.se

Karsten Trostmann ist Expert Director bei CORE. Der Informatiker deckt folgende Schwerpunktthemen ab: IT-Strategie, Evolutionäre IT-Architektur, Domain Driven Design, agile Softwareentwicklung und Cloud Infrastrukturen. Karsten's Erfahrungen umfassen unter anderem die Evaluierung der Plattformstrategie für einen Digitalversicherer, die Entwicklung Cloud Operations für einen Identityprovider, Architekturreviews in verschiedenen Projekten.

Mail: karsten.trostmann@core.se

CORE SE
Am Sandwerder 21-23
14109 Berlin | Germany
https://core.se/
Phone: +49 30 263 440 20
office@core.se

COREtransform GmbH
Am Sandwerder 21-23
14109 Berlin | Germany
https://core.se/
Phone: +49 30 263 440 20
office@core.se

COREtransform GmbH
Limmatquai 1
8001 Zürich | Helvetia
https://core.se/
Phone: +41 44 261 0143
office@core.se

COREtransform Ltd.
Canary Wharf, One Canada Square
London E14 5DY | Great Britain
https://core.se/
Phone: +44 20 328 563 61
office@core.se

COREtransform Consulting MEA Ltd.
DIFC – 105, Currency
House, Tower 1
P.O. Box 506656
Dubai I UAE Emirates
https://core.se/
Phone: +97 14 323 0633
office@core.se