

Von den Grossen lernen

Betriebliche Informationssysteme werden immer vielschichtiger, vernetzter und umfangreicher. Unsere gängigen Methoden damit umzugehen, stossen an ihre Grenzen. Es gibt jedoch Hilfe von denjenigen, die für die Konstruktion von Grossanlagen und Megastrukturen verantwortlich sind. Da können wir was lernen.



«**Systems Engineering hilft durch die klare Abgrenzung von Komplexität und Kompliziertheit.**»

Daniel Liebhart ist Dozent für Informatik an der ZHAW (Zürcher Hochschule für Angewandte Wissenschaften) und Solution Manager der Trivadis AG.

Grossanlagen wie Vertigungsstrassen oder Kraftwerke und Megastrukturen wie Flughäfen, Raumstationen oder Flugzeugträger sind allesamt sehr umfangreiche Vorhaben, die nur auf Basis einer umfassenden Methodik umgesetzt werden können. Diese Methodik wird «Systems Engineering» genannt, und es existiert mit der INCOSE (International Council on Systems Engineering) eine Organisation, die seit einem halben Jahr das Systems Engineering Body of Knowledge (SEBoK) – entsprechend unserem SWEBoK (Software Engineering Body of Knowledge, von der IEEE) – als «public WIKI» betreibt (www.sebokwiki.org). Darin werden die Grundlagen des Systems Engineerings in sieben sogenannten Knowledge Areas beschrieben. Und wir finden darin sehr viel Nützliches zum Thema komplexe Gesamtsysteme, beispielsweise den Systembegriff, der sich sehr gut in einer IT-Landschaft eines Unternehmens anwenden lässt. Und wir finden Prozessstandards für die Umsetzung und Tipps zum Umgang mit Komplexität und anderen Phänomenen.

Systeme und Software

Die INCOSE definiert Systems Engineering als interdisziplinärer Ansatz, der es erlaubt, erfolgreich Systeme zu realisieren. Während wir das Wort System gern für Software, Hardware oder für eine Kombination von beiden verwenden, ohne es genau abzugrenzen, muss es der Systems Engineer ganz genau wissen. Da wird das System als eine Kombination von interagierenden Elementen, das einem definierten Zweck zu erfüllen hat, verstanden. Dies umfasst Hardware, Software, Firmware, Menschen, Informa-

tion, Techniken, Einrichtungen, Dienste und andere Hilfsmittel. Während wir dazu neigen, den Menschen als ausserhalb des Systems zu betrachten und unsere Systeme so auszulegen, dass sie möglichst flexibel und fehlerresistent sind, sieht der Systems Engineer den Menschen als integralen Bestandteil des Systems, genau wie Informationen oder Techniken. Wenn ein Bestandteil des Systems nicht richtig funktioniert, funktioniert das System nicht.

Prozessstandards

Für den Bau umfangreicher Gesamtsysteme existiert eine Reihe von Standards. Neben dem SEBoK von INCOSE haben ISO/IEC, DOD, IEEE und SEI solche Standards veröffentlicht. Sie enthalten Systems-Engineering-Prozesse, die aufgrund der Aufgabenstellung (dem Bau grosser Systeme) notwendigerweise sehr umfangreich sind. Interessant ist für unsere Branche die Gesamtsicht, die unseren Software-Engineering-Prozessen oftmals fehlt. So pflegen wir beispielsweise in den Phasen «PLAN – BUILD – RUN» als Lebenszyklus eines Informationssystems zu denken. Allenfalls wird noch ein DECOMMIT als Schlussphase vorgesehen. Der System Lifecycle Process von ISO/IEC 15288 umfasst zusätzlich Phasen wie «Operational Test & Evaluation» und «Transition to Operation». Beides Aspekte, die wir beim Bau von Systemen oftmals unterschätzen. Für den Bau und die Pflege umfangreicher IT-Landschaften bieten die INCOSE-Systems-Engineering-Prozesse einen wahren Fundus von Erkenntnissen, die schon längst ihren Platz in unserer täglichen Arbeit haben sollten, wie die Erkenntnis, dass

Anforderungen nicht nur systematisch aufgenommen werden sollten (darin sind wir langsam richtig gut), nein, sie sollten auch gegen mögliche Lösungen validiert und allenfalls abgelehnt werden. Oder auch die Differenzierung zwischen Technical Management, bestehend aus Planung, Assessment und Kontrolle, und Technical Evaluation, bestehend aus Analyse, Validierung und Verifikation. Eine Differenzierung, von der wir noch weit entfernt sind.

Komplexität

Ein weiterer wichtiger Aspekt unserer betrieblichen Informationssysteme ist deren Komplexität. Und die Mittel, wie wir mit Komplexität umzugehen versuchen. Auch da hilft Systems Engineering weiter. Zum Beispiel durch eine klare Abgrenzung zwischen Komplexität und Kompliziertheit. Kompliziert ist etwas, was wir nicht verstehen. Der Systems Engineer löst komplizierte Probleme durch Zerlegung in Teilprobleme, die einzeln zu übersehen und damit zu lösen sind. Komplex ist ein System ist aus Sicht des Systems Engineerings dann, wenn es nicht zerlegbar ist und es sich in einer grossen Anzahl von Zuständen befinden kann, die durch regelgeleitete und rekursive Verhaltensmuster definiert sind. Ob Komplexität überhaupt reduziert werden kann, ist umstritten, da sie oftmals durch den funktionalen Umfang eines Systems bedingt ist. Eine Tatsache ist jedoch, dass unsere Informationssysteme in der Komplexität zunehmen und dass die unterschätzte Komplexität ein Grund für unsere Schwierigkeiten in der Pflege bestehender Systeme und in der Integration neuer Bestandteile darstellt. Grossanlagen und Megastrukturen sind per se fast immer komplex. Und im Standardwerk des Systems Engineerings von Alexander Kossiakoff steht es klipp und klar: «Die Funktion von Systems Engineering ist das Anleiten der Konstruktion komplexer Systeme». Die IT von morgen wird nicht einfacher werden – Zeit, schon heute von den Grossen zu lernen. <