

SPiCE und RPA

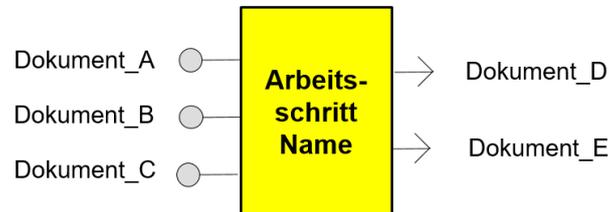
Was haben SPiCE und RPA gemeinsam? Die Lösung ist ganz einfach; Das „P“! Natürlich ist nicht der Buchstabe „P“ gemeint, sondern das, wofür er steht: „Prozess“. SPiCE (Software Process improvement and Capability determination) führt zu einer dokumentierten Festlegung der Softwareentwicklungsprozesse und RPA (Robotic Process Automation) befasst sich damit, wie datenverarbeitende Prozesse mittels Software-Roboter automatisiert werden können. Was ist naheliegender als beides zu RSPA zu kombinieren?



Der internationale Standard SPiCE (ISO/IEC 15504) kommt immer dann zum Einsatz, wenn die Qualität einer zu entwickelnden Software hohe Priorität hat. Meist handelt es sich dabei um Software für Embedded Systeme und meist steht auch das Schlagwort „Funktionale Sicherheit“ im Raum. Typische Anwendungsbeispiele sind die Implementierung von Steuergeräten für die Bereiche „Automotive“, „Luft- und Raumfahrt“ oder allgemein „industrielle Produktion“ im Sinne von Industrie 4.0 . Der grundlegende Ansatz von SPiCE setzt voraus, dass im Unternehmen dokumentiert ist „wie“ die Software zu produzieren ist. D.h. es ist festgelegt welche Prozesse bei der Softwareentwicklung durchzuführen sind und insbesondere welche Dokumente dabei mit welchem Inhalt zu erstellen sind. Die Summe dieser Festlegungen bildet implizit ein Modell der real durchzuführenden Prozesse und SPiCE sieht nicht nur eine Methode vor, wie die Abweichungen von diese Modell zum realen Produktion „gemessen“ und bewertet werden können, sondern definiert auch explizit Mindestanforderungen an die durchzuführenden Prozesse. Diese Mindestanforderungen sind die Quintessenz aus zahllosen untersuchten, erfolgreich durchgeführten Softwareprojekten und es gilt die Hypothese: „Je mehr dieser Mindestanforderungen eingehalten werden, desto besser wird die Qualität der Software sein!“

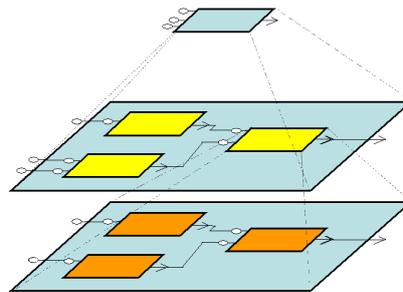
Der Grundbaustein eines Unternehmensmodells ist der „Prozess“. SPiCE und RPA befassen sich mit Prozessen in denen Dokumente oder allgemeiner betrachtet Dateien und Daten-Streams verarbeitet bzw. produziert werden. Solche Prozesse sind dadurch charakterisiert, dass in einem elementaren Arbeitsschritt, aus einen Satz von Eingangsdokumenten, nach einer bestimmten Vorschrift, genau ein Ausgangedokument produziert wird. Da das Erfassen der relevanten Daten aus den

Eingangsdokumenten in der Regel sehr aufwendig ist, werden in der Praxis, aus Gründen der Effizienz, mehrere elementare Arbeitsschritte mit demselben Set an Eingangsdokumenten, zu einem nicht-elementaren Arbeitsschritt mit mehreren Ausgangsdokumenten zusammengefasst. Ein Arbeitsschritt kann somit grafisch wie folgt dargestellt werden.



Schematische Darstellung eines Arbeitsschrittes mit Eingangs- und Ausgangsdokumenten.

Da die in einem Arbeitsschritt produzierten Ausgangsdokumente in der Regel nur Zwischenprodukte sind und als Eingangsdokumente anderer Arbeitsschritte weiterverarbeitet werden, kann ein Prozess als verkettetes Netzwerk von Arbeitsschritten gesehen werden, wobei es in der Regel nur noch eine hierarchische Strukturierung in zusammengesetzte Prozesse ermöglicht, den Überblick zu behalten.



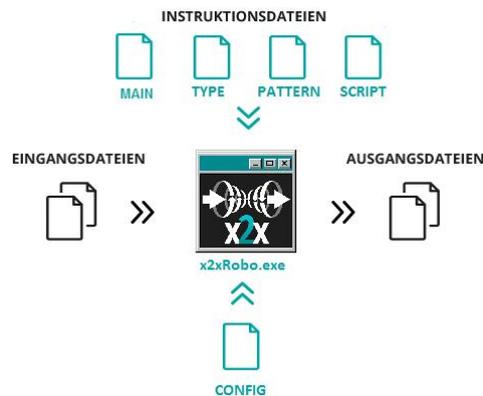
Schematische Darstellung hierarchisch strukturierter Prozesse bzw. Arbeitsschritte.

Einen Arbeitsschritt zu beschreiben, d.h. mehr oder weniger formal festlegen, ist leichter gesagt als getan und man gerät hierbei schnell in einen „Sumpf“, wenn man keine Prozessbeschreibung zur Hand hat, die festlegt, wie das zu tun ist. So – und jetzt sind wir beim „Henne-Ei-Problem“! Wir brauchen einen Prozess in dem festgelegt ist, wie Prozesse zu beschreiben sind. SPiCE definiert dazu den Prozess „SUP.7 Documentation“ – allerdings nur so rudimentär, das man erst einmal nicht weiß wie das konkret gehen soll. Hier hilft uns jetzt ein Blick auf RPA weiter.

RPA (Robotic Process Automation) sieht vor, dass ein Prozess bzw. im einfachsten Fall genau ein elementarer Arbeitsschritt von einer Software automatisiert durchgeführt wird. D.h. in einer solchen Software muss bis in letzte Detail genau festgelegt sein, wo die relevanten Daten in den Eingangsdokumenten zu finden sind und wo die aufbereiteten Daten in dem Ausgangsdokument zu platzieren sind.

Betrachten wir dazu den Fall, dass die Eingangs- und Ausgangsdokumente Formulare im ursprünglichen Sinne sind. D.h. sie bestehen aus einer Abfolge von ausgefüllten oder noch auszufüllenden Eingabefeldern, die mit einem vorausgehenden Titel versehen sind. (Denken sie an den klassischen Fall der Adressenerfassung: Vorname: xxx, Name: xxx, Ort: xxx, ...) In diesen einfachen Fall können wir die Beschreibung des Arbeitsschrittes in drei Teile zerlegen. Wir haben ein

noch nicht ausgefülltes Formular für ein Ausgangsdokument mit fest vorgegebenem Text und benannten leeren Feldern (sogenannte Platzhalter), in die etwas eingetragen wird. Dieses (noch) nicht ausgefüllte Formular hat also die Funktion einer „Dokumentvorlage“. So wird das Ausgangsdokument grundsätzlich aussehen. Um die relevanten Daten aus einem Eingangsdokument zu erhalten, verwenden wir ein entsprechendes leeres Formular als Schablone, mit benannten „Löchern“ (Variablen) drinnen, die über das Eingangsdokument gelegt wird. D.h. eine solche Dokumentschablone legt genau fest, wo die relevanten Daten in einem Eingangsdokument zu finden sind und weist ihnen entsprechende Namen zu. Der noch fehlende dritte Teil unserer formalen Arbeitsschrittbeschreibung besteht dann aus einer Folge von Anweisungen der Art: „Weise den Wert des Platzhalters X den Wert der Variablen Y zu.“



Für den X2X Software-Roboter bilden „PATTERN“ die Dokumentvorlagen für Ausgangsdateien und ein „TYPE“ das „Formular“, das beschreibt „wo“ die Daten in Eingangsdokumenten gefunden werden können. Sonstige Anweisungen zur Durchführung eines Arbeitsschrittes werden in „MAN-“, „SCRIPIT-“ und „CONFIG-Dateien“ festgelegt.

Zurück zu SPICE. Ein wesentlicher Bestandteil von Prozessbeschreibungen sind die Dokumentvorlagen. In einer Dokumentvorlage ist für eine bestimmte Art von Ausgangsdokumenten soweit als möglich festgelegt, wie das zu erstellende Dokument auszusehen hat bzw. was es beinhalten muss. Dabei hat es sich in der Praxis bewährt, ein Ausgangsdokument, wie bei einem Formular, aus einer Kopie der Dokumentvorlage entstehen zu lassen, wobei allerdings deren Inhalt, z.B. durch Verwendung spezieller Schriftarten, vom konkreten Dokument abgesetzt wird.

Die Norm SPICE ist so formuliert, als würden alle Arbeitsschritte / Prozesse von einem Menschen ausgeführt werden, d.h. Prozessbeschreibungen und Dokumentvorlagen sind nach den Bedürfnissen und Möglichkeiten des Menschen formuliert. RPA dagegen setzt auf den Software-Roboter. Mit dem assoziativen Denken eines Menschen kommt ein Software-Roboter grundsätzlich nicht zurecht, (auch wenn KI manchmal etwas anderes vorgaukelt.) Ein Software-Roboter braucht strenge formale Vorgaben.

Nun sind wir bei „des Pudels Kern“ angelangt. Wir haben auf der eine Seite Prozesse, die von Menschen gelebt werden und die für Menschen gemacht wurden, und auf der anderen Seite den Software-Roboter der uns bei der Arbeit helfen soll und deshalb in diese Prozesse integriert werden muss, aber eine ganz andere „Sprache spricht“. Wie soll das gehen?

Wir haben Glück – Dokumente sind Dateien – und Software kann Dateien lesen. Das heißt, die entscheidenden Schnittstellen zwischen Mensch und Maschine sind solche Dateien, die der Mensch erzeugen und die Maschine lesen oder umgekehrt die Maschine erzeugen und der Mensch lesen kann. Genau dann, wenn alle Eingangsdokumente eines Arbeitsschrittes von der Maschine gelesen werden können, kann dieser Arbeitsschritt von einem Software-Roboter vollautomatisch durchgeführt werden. Dass, dieser Arbeitsschritt für den Software-Roboter in der Regel komplett anders beschrieben werden muss, als für einen Menschen, ist dabei ebenso selbstverständlich, wie unbedeutend. Entscheidend ist einzig und allein, dass die von einem Menschen erstellten Eingangsdokumente, auch für die Maschine lesbar sein müssen. – Und das richtet sich danach, wie „schlau“ die Maschine ist und welche Zugeständnisse der Mensch dabei machen kann. Aufgrund der Leistungsfähigkeit heutiger und zukünftiger Computer, können wir aber davon ausgehen, dass auch die Maschine hierbei dem Menschen immer mehr entgegen kommt. Das erklärt auch die schier grenzenlosen Erwartungen an RPA, bei denen ein Software-Roboter schnell mal mit einem Androiden gleichgesetzt wird. Aber noch sind wir nicht soweit!

Was ist eigentlich ein Software-Roboter?

Ein sehr einfache, aber um es vorwegzunehmen, ungenügende Antwort darauf ist: „Ein Software-Roboter ist eine Software, die den Menschen bei der Verarbeitung von Dokumenten ersetzen könnte.“ Irrtum! Dieses Bild vom Software-Roboter als Androiden ist falsch. In den derzeitigen realen Produktionsprozessen gibt es keine Arbeitsschritte bei deren Durchführung der Mensch nicht assoziativ und kreativ Wissen verknüpft um dadurch sinnvolle Ergebnisse zu produzieren. Dieses Wissen ist aber nicht in Dokumenten abgelegt; mithin einer Maschine nicht zugänglich. Somit der Mensch auch nicht unmittelbar ersetzbar.

Wie weiter oben bereits erläutert, müssen alle Informationen die die Maschine braucht, um einen Arbeitsschritt selbständig und automatisch durchführen zu können, als maschinenlesbare Dokumente vorliegen. Das heißt, ein für den Menschen geschaffener Arbeitsschritt muss erst so in mehrere Arbeitsschritte aufgespalten oder umgewandelt werden, dass wenigstens ein Teil dieser reduzierten Arbeitsschritte die Voraussetzung zu Automatisierung erfüllt. Diese können dann tatsächlich durch die Maschine vollautomatisch durchgeführt werden, sind aber keine Arbeitsschritte mehr, bei denen sie den Menschen ersetzen, sondern solche die extra für die Maschine geschaffen wurden. Dieses Aufspalten, Umwandeln und wieder Zusammensetzen von Arbeitsschritten ist ein wesentlicher Schritte beim Übergang von der normalen (Software-)Produktion zur teilautomatisierten (Software-)Produktion mittels „Robotic Process Automation“ (RPA).

Zweiter Versuch: „Ein Software-Roboter ist eine Software die vollautomatisch nach einer bestimmten Vorschrift Eingangsdateien / -dokumente/ -datenstreams in Ausgangsdateien/ -dokumente/ -datenstreams umsetzen kann, wobei die Vorschrift austauschbar als Datensatz außerhalb der Software hinterlegbar sein muss.“ Der letzte Abschnitt dieser Definition ist absolut von Bedeutung und entscheidet darüber, wann gewöhnliche „Process Automation“ das so viel versprechende Attribut „Robotic“ verdient.

Eine Automatisierungssoftware zu erstellen bringt alle Nachteile mit sich, die das Erstellen von Software im Allgemeinen beinhaltet. Er ist zeit- und kostenintensiv und dauert Monate oder Jahre, wenn es am Ende nicht sogar unvollendet abgebrochen wird. Bei der wirtschaftlichen Betrachtung

von Prozessautomatisierung ist immer der ökonomische Nutzen aus dem automatisierten Arbeitsschritt den Kosten für die Realisierung der Automatisierungssoftware gegenüberzustellen. Tja, und wenn man hierzu immer wieder, von Grund auf ein Softwareprojekt zur Erstellung der Automatisierungssoftware starten muss, dann wird das nur selten geschehen. Der Ausweg aus diesem Dilemma ist der Software-Roboter. Für diese instruierbare Automatisierungssoftware ist kein Schreiben, Ändern oder Anpassen von Sourcecode notwendig und somit keine unwägbare Wartezeit auf das Ergebnis, wie allgemein bei Softwareentwicklung üblich.

In der Tat! Das Instruieren eines Software-Roboters spielt sich in der Praxis im Bereich von Stunden oder maximal wenigen Tagen ab. Das ist die Magie hinter „Robotic Process Automation“: In kürzester Zeit und mit wenig Aufwand ein Ergebnis zu erhalten, von dem man viel und lange profitiert. Die Einführung von SPiCE und der damit verbundene Aufwand Prozesse zu erfassen und dokumentieren kann also durchaus sehr lohnenswert mit dem Übergang zu RPA verknüpft werden.

Weiterführende Dokumente:

[http://www.sss.de/files/triple-s/downloads/downloads_sonstiaes/Einfuehrung%20SPiCE%20Kapitel%201-3%20\(Josef%20Huebl\).pdf](http://www.sss.de/files/triple-s/downloads/downloads_sonstiaes/Einfuehrung%20SPiCE%20Kapitel%201-3%20(Josef%20Huebl).pdf)

http://www.sss.de/files/triple-s/downloads/downloads_X2X_2015/Robotic%20Process%20Automation.pdf

http://www.sss.de/x2x-downloads?file=files/triple-s/downloads/downloads_X2X_2015/Sourcecode_generieren.pdf

http://www.sss.de/x2x-downloads?file=files/triple-s/downloads/downloads_X2X_2015/Modellbasierte%20Softwareentwicklung.pdf

http://www.sss.de/x2x-downloads?file=files/triple-s/downloads/downloads_X2X_2015/Funktionale_Sicherheit.pdf

<http://x2x.sss.de>