

Call for Papers – Beitrag für den Kongress DASMA / MetriKon 15./16. November 2007

SOA & Metriken – die Aufwandsschätzung –

Einsatz der Function-Point-Methode in einem durchgängigen SOA-Entwicklungsprozess

Dierk Pingel

Kurzbeschreibung des Inhaltes

Beim Betrieb von Service-Orientierten Architekturen (SOA) kommen in der Regel Service-Busse zum Einsatz, die mit ihren Prozess-Engines fachliche Prozesse steuern und überwachen. Im SOA-Umfeld ist dafür der Name Business Activity Monitoring (BAM) geprägt worden. Mit den gewonnenen Metriken können z.B. Schwachstellen in der Prozessbearbeitung identifiziert und Verbesserungsvorschläge erarbeitet werden. Das ist nicht neu. Schon in den 90-er-Jahren wurden Metriken für die Geschäftsprozessverarbeitung benutzt.

Auch Metriken für die Systementwicklung, wie z.B. die Aufwandsschätzungen von Entwicklungsarbeiten, die Qualität von Software, die Produktivität der Softwareentwicklung sind nicht neu. So erhebt beispielsweise die Function-Point-Methode den Anspruch, methodenneutral und damit auch für SOA-Anwendungen einsetzbar zu sein.

In dem Beitrag soll gezeigt werden, wie die Function-Point-Methode bei der Entwicklung von SOA-Anwendungen verwendet werden kann. Dazu wird in aller Kürze ein durchgängiger SOA-Entwicklungsprozess beschrieben (eine längere Fassung steht in [Pingel 2007]) und die Verbindung zu der Function-Point-Zählung hergestellt.

Eine Schwäche der Function-Point-Methode ist die Bestimmung der Elementarprozesse, die Basis für die Zählung der EIs, EOs und EQs (external inputs, outputs und inquiries) sind. [Garmus 1996, S. 55] schreibt „An external input [EI] is an elementary process of the application which processes data or control information that enters from outside the boundary of the application.“ und „The transactional functions, external inputs (EIs) ...perform the processes of ... retrieval, ... (transactions you would

expect to see in a process model).“. Mit Blick auf die Methodenneutralität wird nicht weiter darauf eingegangen, wie das Prozessmodell entsteht.

Wenn eine Applikation und damit das Prozessmodell bereits vorhanden ist, ist das kein Problem. Die Function-Point-Methode erhebt allerdings den Anspruch, auch schon in einer sehr frühen Phase der Softwareentwicklung zählen zu können, so dass in irgendeiner Weise ein Prozessmodell entstehen muss.

Das Erstellen eines Prozessmodells ist immer auch Teil eines Entwicklungsprozesses, so dass hier Synergien zwischen dem Entwicklungsprozess und der Function-Point-Methode nicht nur entstehen, sondern zwingend benötigt werden. Ein Auseinanderlaufen des „Function-Point-Prozessmodells“ und des „Entwicklungsprozessmodells“ kann nur vermieden werden, wenn es nur ein Prozessmodell gibt, das gleichermaßen für die Entwicklungsarbeiten und für die Function-Point-Zählung benutzt wird. Die Function-Point-Methode sollte daher nicht als eigenständige Methode betrachtet werden, sondern als Ergänzung der ohnehin notwendigen Methoden für einen Softwareentwicklungsprozess.

Dieser Beitrag wird zeigen, wie die Function-Point-Methode einen in [Pingel 2007] beschriebenen durchgängigen SOA-Entwicklungsprozess ergänzt.

Viele Beispiele im Function-Point-Umfeld verwenden für die Zählung technische Artefakte, wie z.B. Bildschirmmasken, die aber erst im Design der Anwendung entstehen werden und zu Beginn der Softwareentwicklung noch gar nicht vorliegen können. Es wird in diesem Beitrag daher auch darauf eingegangen, wann organisatorische Artefakte, z.B. Arbeitsanweisungen, entstehen und ob bzw. welchen Einfluß sie auf die Function-Point-Zählung haben sollten.

Die Aufwandsschätzung verwendet die Function-Points als virtuelles Maß für den Umfang einer Applikation. Mit Hilfe von Erfahrungswerten (-datenbanken) oder mit Hilfe von Berechnungsformeln, die aus Erfahrungswerten abgeleitet werden, wird ein Aufwand für die Applikation bestimmt. Ein Korrekturfaktor bei der Zählung der Function-Points in Form von 14 Einflußfaktoren und Korrekturwerte bei der Umsetzung in eine Aufwandsschätzung sollen helfen, die Aufwandsschätzung genauer werden zu lassen.

Man weiß, dass nicht nur Technologien und die Soft- und Hardskills von Menschen den Aufwand beeinflussen, sondern auch der Entwicklungsprozess selbst. Durch die engere Bindung der Function-Point-Methode an einen Entwicklungsprozess kann die Einflußgröße ‚Entwicklungsprozess‘ an Bedeutung verlieren und damit die Aufwandsschätzung genauer werden.

Ein guter Entwicklungsprozess sollte neben den methodischen auch organisatorische Vorgaben enthalten, wie z.B. die Zusammensetzung eines Projektteams und die Zusammenarbeit innerhalb des Projektes. Dann ließe sich auch der Einflußfaktor ‚Mensch‘ bei der Aufwandsschätzung enger fassen.

Der Einflußfaktor ‚Technologie‘ adressiert zum einen den Technologieeinsatz bei der Entwicklungsarbeit selbst (z.B. eingesetzte Methoden und Werkzeuge) und zum anderen die Implementationsplattform, in der die Applikation realisiert wird (z.B. ablauforganisatorische Umsetzung, EDV-Umsetzung in Form von Host-Programmen oder Office-Anwendungen). Durch die Trennung des SOA-Entwicklungsprozesses in technologieunabhängige und –abhängige Phasen und durch den Einsatz von Generatoren bei der technischen Umsetzung verliert auch die ‚Technologie‘ an Einfluß auf die Aufwandsschätzung. Die Aufwandsschätzung kann dadurch wiederum genauer werden.

Die Anbindung der Function-Point-Methode an den vorgestellten SOA-Entwicklungsprozess löst auch das Problem der Granularität eines Elementarprozesses. Für Services im SOA-Umfeld gilt vergleichbares. Viele Entwicklungsprozesse lassen offen, wie man zu der „richtigen“ Granularität von Services kommt. Der vorgestellte SOA-Entwicklungsprozess führt den Entwickler „automatisch“ zu **einer** „richtigen“ Granularität. Elementarprozesse und der durch den SOA-Entwicklungsprozess geprägte Ausdruck des Services-Prozesses werden gleichgesetzt, so dass das „Problem“ der Granularität von Elementarprozessen durch den SOA-Entwicklungsprozess mit gelöst werden kann.

Der vorgestellte SOA-Entwicklungsprozess wird erst im Design SOA-spezifisch. Er kann somit zusammen mit der Function-Point-Methode in fast allen, beliebig großen, kommerziellen Projekten eingesetzt werden.

Literatur

- [Garmus 1996] Garmus, D., Herron, D., Measuring the Software Process, Yourdon Press, Prentice Hall, Upper Saddle River, New Jersey, ISDN 0-13-349002-5
- [Pingel 2007] Pingel, D., Der SOA-Entwicklungsprozess auf Seite 187 ff. in Starke, G., Tilkov, S., SOA-Expertenwissen, dpunkt.verlag, Heidelberg, ISBN 978-3-89864-437-2