

Das Softwareprojekt war erfolgreich! Sind Sie ganz sicher?

Autor : Bogdan Lent

Datum : 1. September 2017



Hohe Softwareunterhaltskosten können mit dem Einsatz von Komplexitätsmetriken gesenkt werden. Eine Berücksichtigung der entsprechenden Anforderungen bereits bei der Beschaffung von Software wirkt nachhaltig. Möglich sind zwei Verfahren, beide mit ihren Vor- und Nachteilen, abhängig von der Strategie und der Investitionsbereitschaft des Beschaffers. Es wird der langfristige Aufbau der entsprechenden Kompetenzen und Messsysteme seitens der Beschaffer empfohlen.

Bisherige Beschaffungspraxis bei Investitionen für Individualsoftware

Technical Debts (Technische Schulden, also Pendenzen aus dem Entwicklungsprozess, resultieren meistens wegen noch nicht erstellten Teilen, welche die funktionalen Anforderungen zu erfüllen haben), werden weitgehend durch bewusste Managemententscheide unter Inkaufnahme unerwünschter Konsequenzen verursacht. Die Unterhaltskosten, welche den Technical Debts zuzuschreiben sind, sind vor allem durch die Qualität der beschafften Software beeinflusst. Ein jährlich steigender Aufwand um 15-20% der Investitionskosten ist mittlerweile der allgemeine Industriestandard [1]. Können wir also von einem erfolgreichen Softwareprojekt sprechen, wenn es nach allen Regeln der Kunst auf Termin, im Rahmen der vereinbarten

Budgets realisiert wurde und gar die verlangten Funktionalitäten weitestgehend aufweist, aber technische Schulden und künftige Unterhaltskosten ausser Acht lässt?

Bisher standen bei der Erstellung von Beschaffungs- und Abnahmekriterien typischerweise die funktionalen und nichtfunktionalen Anforderungen im Fokus. Weniger Gewicht wurde auf vermeintlich unbedeutende Eigenschaften wie z.B. Architektur, Kommentare oder Dokumentation der Software gelegt. Die Lösung der Aufgaben, ggf. deren „Verkomplizierung“ durch unklare und unnötig komplizierte Strukturen, ist weitestgehend in der Hand der Hersteller [2]. Das Problem unnötig komplizierter Software kennt man übrigens nicht nur bei so genannter Legacy- und monolithischer Software, gerade die hochgelobten agilen Techniken fördern schlechte Lösungsqualität: egal wie, wichtig ist vor allem, dass der Kunde schnell zu einem funktionierenden Produkt kommt, irritierenderweise oft auch «Minimum viable product» (MVP) genannt, was so ziemlich das Gegenteil vom wirklichen MVP («Most valuable player») ist. Der Kunde ist mit der Erfüllung der Funktionen mit akzeptablen Reaktionszeiten zufrieden – der Abnahme steht nichts mehr im Weg[3]. Nur leider wird den im nachmaligen Betrieb entstehenden Folgekosten zu diesem Zeitpunkt keine oder kaum Aufmerksamkeit geschenkt. Ausserdem hat der Lieferant der Software damit einen robusten Schuh in der Tür des Beschaffers für lukrative Folgeaufträge (Selbstverständlich hat die Branche auch hierfür klingende und eindruckliche Begriffe wie Architectural Redesign, Modularisierung, Refactoring, Microservices implementieren etc.).

Höchste Zeit also, etwas zu ändern

Die Komplexität (genauer: die Verkomplizierung der Lösung durch die unter Zeitdruck entstandene Software) solcher Produkte lässt sich messen. Vier von acht Qualitätskriterien für gute Software-Entwicklung der Norm ISO 25010 sollten bereits im Ansatz gegen die Verkomplizierung der Software wirken[4]:

Einfache Wartung (modularer Aufbau, wiederverwendbare Komponenten, gute Analyse-Funktion, leicht modifizierbar, umfangreiche Testoptionen)

- Leichte Portierbarkeit (gute Adaptivität, leicht zu installieren, einfach austauschbar)
- Verlässliche Software (ausgereifte Softwarequalität, Verfügbarkeit, Fehlertoleranz, Wiederherstellbarkeit)
- Hohe Kompatibilität (optimale Co-Existenz zu weiterer Software, Interoperabilität)

Die übrigen Kriterien der Norm wie funktionale Software, perfekte Usability, hohe Effizienz und höchste Sicherheit sind zunehmend in den Beschaffungsanforderungen spezifiziert – deren mangelnde Erfüllung kann den Technical Debts zugeordnet werden. Die übrigen Kriterien der Norm wie funktionale Software, perfekte Usability, hohe Effizienz und höchste Sicherheit sind zunehmend in den Beschaffungsanforderungen spezifiziert – deren mangelnde Erfüllung kann den Technical Debts zugeordnet werden.

Zwei der Kriterien: leichte Portierbarkeit und die hohe Kompatibilität wirken sich auch auf Erfüllung der übrigen Kriterien aus, vor allem auf die Sicherheit und auf die Zuverlässigkeit. Softwaresysteme werden nicht selten über 2 -3 Generationen der technologisch bedingten Erneuerungen der Trägersysteme verwendet [5]. Bei einem langjährigen Einsatz werden somit

die Migrationen auf jeweils neuere technologische Plattformen notwendig. Beide Kriterien - die leichte Portierbarkeit und die hohe Kompatibilität - sind durch die Softwarestrukturen, insbesondere so genannte zyklische Abhängigkeiten bedingt.

Jede Komponente einer Software ist mit anderen vernetzt. Je geringer ihre Kopplung, desto besser ist ihre Testbarkeit, Wartbarkeit und Verständlichkeit (Faktoren der oben erwähnten Softwarequalitätskriterien: Einfache Wartung und Verlässliche Software) (Law of Demeter [6]). Diese Kopplung kann mit Kopplungsmetriken wie Normalized Cumulative Component Dependency (NCCD) oder Average Component Dependency (ACD) gemessen werden [7].

Die Kopplung wird vor allem durch die zyklischen Abhängigkeiten bestimmt: je weniger Abhängigkeiten, desto geringere Kopplung. Dies wieder kann mit Messung von Umfang von Ausdrücken nach Halstead [8] oder Anzahl der binären Verzweigungen nach McCabe [9] gemessen werden. Als Faustregel gilt: je höher die Werte der Metriken, desto komplexer und fehleranfälliger ist die Software.

Was kann man dagegen tun? Und zwar gleich am Anfang!

Zunächst die Metriken der Komplexität für ein Unternehmen definieren, die Zielwerte festlegen und diese als Anforderungen im Pflichtenheft dem Lieferanten der Software auferlegen.

Dies reicht allerdings nicht – solche Auflagen können im einfacheren Fall zu Verzögerungen bei der Abnahme und im schlimmsten Fall zum Scheitern eines Projektes führen. Um die Komplexität der Software und damit die dereinstigen Unterhaltskosten nachhaltig zu limitieren, muss diese Art Qualität der zu erstellenden Software bereits in der Beschaffung fest verankert und auf konkrete Werte der gewählten Metriken abgestützt sein. Die Lieferungen müssen nachweislich die Metriken erfüllen.

Allerdings sind die Kosten für den Einsatz, für die Werkzeuge, für die Definition der Metriken und für die Erfassung geeigneter Messwerte während der Tests, nicht unerheblich [10]. Die Automatisierung weist hier noch Wachstums- und Verbesserungspotential auf – bis auf weiteres ist ein substantieller Einsatz von spezialisierter Expertise unerlässlich. Obwohl letztendlich der Beschaffer sämtliche Kosten eines Vorhabens trägt, wünschen sich meistens beide Parteien, die Kosten für die Festlegung dieser Kriterien von der anderen Seite getragen zu sehen, unabhängig davon, ob beide Seiten sachlich in der Verwendung der Metriken übereinstimmen.

In der Praxis kann eines von zwei Verfahren angewendet werden:

1. Dem Lieferanten wird die Auflage gemacht, die volle Kostenträgerschaft für die Beschaffung der Messsysteme und der Erstellung der Metriken zu übernehmen.

Der Beschaffer definiert die Metriken und die zu erreichenden Werte und verlangt ein Zertifikat der Erfüllung bei der Lieferung. Diese Verfahren sind möglich, wenn:

- ein zertifiziertes, nicht beeinflussbares Messsystem für die ausgewählte Programmiersprache durch Dritte angeboten wird, und wenn
- beide Parteien (der Beschaffer und der Softwarehersteller) die resultierenden

Prüfergebnisse bedingungslos anerkennen und akzeptieren

Dem Ersteller steht frei, ob er die Aufwendungen weiter in seiner Offerte verrechnet oder nicht. Diese Variante ist wirtschaftlich attraktiv für den Softwarehersteller, wenn er sich strategisch zur Verwendung der Softwaremetriken über breitere Bereiche seiner Produkte entschliesst und durch eine Modularität und hohe Weiterverwendbarkeit die Gesamtkosten des Aufwandes für alle seine Produkte minimieren kann. In einem anderen Fall wird der Lieferant sich scheuen, die hohen Investitionen vor dem Erhalt eines Zuschlags zu tragen. Somit binden sich beide Seiten im Falle eines Zuschlags vertraglich im Glauben und in der Erwartung, die vereinbarten Werte der Metriken erreichen zu können. Dies bedeutet ein Risiko von Nachverhandlungen – welche aber geringere Folgen als das Risiko der minderen Softwarequalität nach sich ziehen.

Sollte der Hersteller aber bereits über ein Messsystem für Softwaremetriken verfügen und dieses einsetzen, liegt es am Beschaffer, über dessen Akzeptanz in Abweichung vom geforderten Messsystem zu entscheiden.

Für den Beschaffer ist dieses Verfahren sicher die günstigere Lösung, weil er nur die vereinbarten Kosten der Zertifizierung zu tragen hat. Erst bei der Lieferung sieht er die Zertifikate, für deren Auswertung keine hohen Fachkompetenzen erforderlich sind.

2. Der Beschaffer beschafft die Messeinrichtungen selbst und führt die Messungen selber durch, resp. beauftragt Dritte mit deren Durchführung während des gesamten Prozesses der Softwareerstellung.

In diesem Fall erwirbt der Beschaffer die Kompetenzen zur Beherrschung des Messsystems, dessen Einsatz und die Möglichkeit, die Programmierfähigkeiten eines Softwarelieferanten noch vor der Erteilung des Zuschlags einem Test zu unterziehen.

Voraussetzungen sind:

- Der Beschaffer verfügt über Kompetenzen der Messsystemanwendung, insbesondere im Bereich der noch nicht automatisierten Einsätze von Experten
- Der Hersteller anerkennt und akzeptiert bedingungslos die Prüfergebnisse des Messsystems, inklusive der Expertenkompetenzen des Beschaffers

Bei diesem Verfahren können die Risiken beider Seiten durch eine Präqualifikation der interessierten Anbieter während des Beschaffungsprozesses wesentlich reduziert werden.

Vom Anbieter wird anlässlich der Offertstellung ein Sourcecode-Muster mit eingeschränkter Funktionalität verlangt. Die Software aller Anbieter wird dem Komplexitätstest unterzogen und die Ergebnisse fliessen in die Auswertung der Offerten ein. Ein Verfahren zur Einschränkung der Anzahl von zu testender Software könnte ebenfalls definiert werden.

In einem zweiten Schritt wird von den Anbietern auf der Shortlist die Umsetzung von für alle gleichen Funktionalitäten unter strengen Umsetzungszeitvorgaben verlangt. Erneut wird die Komplexität der so unter Zeitdruck und unter Beobachtung durch den Beschaffer, erstellten

Software gemessen.

Anschliessend werden die Messwerte für jeden Anbieter nach einem vordefinierten Schlüssel aggregiert und verglichen. Der Anbieter der Software mit den niedrigsten Komplexitätswerten bekommt die höchste Punktezahl für den Zuschlag.

Tabelle 1 stellt die beiden Varianten aus der Sicht des Beschaffers gegenüber.

Nr. Charakteristika