



Softwarewartung – das Mauerblümchen

Viele trendige Themen der Informatik beschäftigen sich mit der Softwareentwicklung auf der „grünen Wiese“. Daneben erscheint die Softwarewartung wenig glanzvoll – wegen der zunehmenden wirtschaftlichen Bedeutung steigt das Interesse aber stetig.

Wartung ist mehr als nur „Fehler flicken“

Haben sie sich schon mal gefragt warum Softwarewartung nötig ist? Schliesslich erfährt Software im Gegensatz zu anderen technischen Systemen weder Abnutzung noch Verschleiss. Und trotzdem braucht es Softwarewartung.

Der wohl bekannteste Grund ist sicher das Beheben von Fehlern. Denn trotz intensiver Tests bleibt eine Anzahl unentdeckter Fehler, sogenannte Restfehler, übrig. Diese offenbaren sich erst nach der Auslieferung mit der Nutzung der Software. Es gibt aber zwei weitere Gründe, die vielleicht weniger bekannt sind.

Eine Software ist nicht isoliert zu betrachten, sondern in eine Umgebung eingebettet, wie beispielsweise eine Hardwareplattform, ein Betriebssystem oder alle möglichen Drittprodukte. Im Laufe der Zeit ändert sich diese Umgebung und erfordert so Anpassungen unserer Software. Unterlässt man diese Anpassungen, veraltet unser System ohne eigenes Zutun, gegenüber der sich verändernden Umgebung. Dieses Phänomen wird im englischen mit „lack of movement“ bezeichnet. Genauso wie das Unterlassen von Änderungen der Software zu deren Alterung führen kann, kann auch das ungeschickte Anpassen zur unnötig schnellen Alterung beitragen. Nimmt ein Entwickler Änderungen vor, ohne die zu Grunde liegenden Architektur- und Designkonzepte zu beachten, werden diese verwässert und führen zu höherer Komplexität der inneren Strukturen. Dies nennt man „ignorant surgery“ und erschwert weitere Anpassungen.

Wartung beginnt in der Entwicklung

Softwarewartung hat somit im Wesentlichen drei Gründe: Restfehler, fehlende Anpassung an die sich verändernde Umgebung und Codeanpassungen durch Missachtung der ursprünglichen Designkonzepte. Mit dem bisher geschriebenen ist klar: Ob Softwarewartung nötig wird, steht schon fest, bevor die erste Zeile Code geschrieben wurde. Wie aufwändig, unangenehm und teuer Wartung wird, steht dagegen noch nicht fest. Darin besteht ein grosser Spielraum, den es massgeblich während der Entwicklung der Software zu nützen gilt. Was aber genau können wir während der Entwicklung tun?

Beginnen wir beim Thema Restfehler. Im ersten Ansatz könnte man denken, mit besseren und umfangreicheren Tests gehören Restfehler der Vergangenheit an. Beim zweiten Blick wird man aber feststellen, dass eine vollständige Testabdeckung überraschend schnell unmöglich wird. Testen ist somit lediglich ein Stichprobenverfahren, welches Auskunft gibt, wie gut man während der Konstruktionsphase der Softwareerstellung gearbeitet hat. Somit werden Restfehler bleiben - wie viele, kann man in der Konstruktionsphase der Softwareentwicklung aber massgeblich beeinflussen.

Hier sind u.a. hilfreiche Massnahmen: eine tragfähige und einfach erweiterbare Architektur, sowie ein übersichtliches Design, das sich Design Patterns zunutze macht. Natürlich sollte man

beides angemessen dokumentieren. Der Entwickler wird es während der Wartung danken und so hoffentlich dem „ignorant surgery“ Problem ausweichen. Natürlich gibt es viele weitere bekannte Massnahmen, wie hohe Kohäsion und lose Kopplung oder das Vermeiden von Codeduplikationen. Vieles davon ist bekannt, geht aber oft in der Hitze des Gefechtes unter und muss in der Wartung teuer bezahlt werden.

Hilfreich für eine spätere Wartung sind aber auch Überlegungen, wie man später Diagnose betreiben will oder neue Software Releases einspielen soll. Welche Mittel hat man später um Fehler schnell aufzuspüren und einzugrenzen? Muss man Software jeweils vor Ort einspielen oder gibt es einen Fernwartungszugang? Kann man ein System für Softwareupdates abschalten oder soll das im laufenden Betrieb möglich sein? Fragen dieser und ähnlicher Art müssen vor Beginn der Softwareentwicklung gestellt und geklärt werden, um später in der Wartung davon profitieren zu können.

Dem Thema „lack of movement“ begegnet man am besten, indem man die Umgebung, in welche die Software eingebettet ist, ständig im Auge behält und Anpassungen an diese bei Erweiterungsarbeiten kontinuierlich berücksichtigt. Hier ist Augenmass gefragt, denn wegen den vielfältigen Abhängigkeiten gibt es auch ein zuviel ohne direkten Kundennutzen.

Softwarewartung ist ein vielseitiges und interessantes Thema, dem es speziell während der Entwicklung genügend Aufmerksamkeit zu schenken gilt, um später davon doppelt und dreifach profitieren zu können. Weitere Einblicke liefern die - bisher noch recht wenigen - Veröffentlichungen (z.B. [1]), die sich diesem Stiefkind der Informatik widmen. (Christoph Bommer/Markus Spindler) ■



Christoph Bommer ist Leiter der Entwicklung der Leittechnik bei Siemens Schweiz AG Mobility



Markus Spindler arbeitet als Quality-manager.

Veranstaltungskalender

Organic - Elektronik der Zukunft

2. Juli 2009, ZHAW Winterthur
Info: www.fael.ch → Anlässe → Focus 542

Telekommunikation (3 Tage Seminar)

1./10./11. September 2009, Berner Fachhochschule, Ilconweg 1, Burgdorf
Info: www.fael.ch → Anlässe → Focus 537

FPGA/ASIC Verifikation mit VHDL

8./9. September 2009, Hochschule für Technik FHNW, Windisch
Info: www.fael.ch → Anlässe → Focus 546