

Modellhafte Verbesserung in einer Testabteilung

White Paper MBT bei Siemens Healthcare MR



MBTsuite
THE TESTING FRAMEWORK



SIEMENS



Download der Testversion:
www.mbtsuite.de



sepp.med gmbh

Geschäftsführer

Dipl. Betriebsw. (FH) Barbara Bilen

Dipl. Inf. Florian Prester

Dipl. Kffr. Maria Prester

Handelsregister Fürth HRB B 6131

Umsatzsteuer-ID: DE 179 168 106

AFRA GmbH

Geschäftsführer

Dipl. Math. Peter Kreutzer

Dipl. Math. Bernhard Nuber

Handelsregister Fürth HR B 3797

Sitz der Gesellschaft: Erlangen

Umsatzsteuer-ID: DE 229 229 489

Impressum: Konzept, Text und Grafik: sepp.med gmbh & AFRA GmbH
Copyright by sepp.med gmbh & AFRA GmbH

Version 1.0

Einführung

Am Ende einer Entwicklung bleibt kaum noch Zeit für den Test. Teure Automatisierungswerkzeuge kommen nicht zum Einsatz, da niemand die Zeit hatte, sich mit deren aufwändigen Konfiguration zu beschäftigen. Alles, was an gründlicher Planung und Testmanagement vorgedacht war, wird durch hastige Last-Minute-Änderungen oder -Ergänzungen überlagert. Dabei wird seit vielen Jahren an verschiedenen Stellen der SW-Entwicklung entweder an der Steigerung der Effektivität oder der Reduzierung des Aufwands gearbeitet. Lassen sich aber unter den gegebenen Rahmenbedingungen gleichzeitig die Effektivität und die Effizienz steigern?

Der konkrete Anwendungsfall

Für die bildgebende Diagnostik mithilfe der Kernspintomografie existieren SW-Pakete für verschiedene Anwendungsfelder:

- Erzeugte Bilder werden durch eine Qualitätssicherung erstmalig auf den Erfolg einer Aufnahme überhaupt geprüft.
- Diese werden anschließend zum Befund durch den jeweiligen Facharzt (z. B. Radiologe, Orthopäde, Urologe etc.) über ein Netzwerk an die entsprechende Fachabteilung geschickt.
- Die Fachärzte nutzen SW-Module, die ihnen durch spezielle Filter oder Einstellungen den fachmännischen Blick auf den zu untersuchenden Patienten erlauben.
- Anschließend werden die gesprochenen diktierten Befunde zusammen mit den Bildern bzw. Bildersequenzen zentral gespeichert.

Solche SW-Pakete existieren innerhalb der Produktfamilie eines Herstellers basierend auf einer proprietären Plattform auch für andere bildgebende Verfahren. Die Anforderungen an die SW sind also multipel und stammen sowohl aus unterschiedlichen Fachabteilungen als auch von deren ebenso unterschiedlichen Kunden. Egal, ob man also an einem Computertomografen, einem Angiographiesystem, einem Ultraschallgerät oder einem Magnetresonanztomografen (sogenannte Modalitäten) von Siemens Healthcare sitzt, immer arbeitet der Benutzer mit einer identischen Benutzeroberfläche.

Diese Software von Siemens „syngo.via“ ist ein Medizinprodukt und muss gemäß den Vorgaben des Medizinproduktegesetzes entwickelt und seine Qualität gesichert werden. So wie das medizinische Diagnostik- oder Therapiegerät, auf dem eine Software installiert ist, ist syngo.via selbst auch ein Medizinprodukt. Durch die EWG 93/42, eine EU-Richtlinie zur Herstellung und dem Vertrieb von Medizinprodukten, darf ein Hersteller diese erst dann in der EU „in den Verkehr bringen“, wenn er nachweislich diverse verpflichtende Normen, die die Entwicklung oder die Herstellung regeln, eingehalten hat. Für die Entwicklung von syngo.via gilt unter anderem die IEC 62304, die z. B. vorschreibt, dass alle Anforderungen an das Medizinprodukt lückenlos durch die Entwicklung und den Test verfolgbar sein müssen. Neben den Fachspezifischen existieren verschiedenste Anforderungen z. B. hinsichtlich der Sicherheit oder der Gebrauchstauglichkeit. Darüber hinaus wird eine neue Version immer auch für alle Medizinprodukte, die diese Software nutzen, freigegeben. Es handelt sich also um eine mehrdimensionale und sehr komplexe Anforderungslandschaft. Über die vergangenen Jahre konnten am kompletten Entwicklungsprozess (in einem V-Modell inkl. der Tests) durch Prozessoptimierung, durch Einführung geeigneter Werkzeuge sowie den Einsatz hoch qualifizierter Mitarbeiter achtbare Erfolge hinsichtlich Bewältigung der ständig steigenden Ansprüche und Anforderungen erzielt werden.

Dabei muss jede weitere Veränderung am Prozess und/oder den Werkzeugen in die bestehende Landschaft eingepasst werden und die betroffenen Prozessteilnehmer dafür gewonnen werden. Je komplexer und größer der Katalog der Anforderungen an die Software ist, desto schwieriger ist es, alle diese einzelnen Komponenten und vor allem deren Integration ins Gesamtsystem detailliert zu testen, um mögliche Fehler herauszufinden und korrigieren zu können.

Das bedeutet viel Arbeit für Stefan Haller und sein Team von Siemens Healthcare, die für den Test von syngo.via im Bereich Magnetresonanztomografie verantwortlich sind: Sie sollen alle nötigen Testfälle beschreiben und durchführen, um zu vermeiden, dass die Software noch Fehler enthält, wenn sie an den Kunden ausgeliefert wird.

Die Beschreibung von Testfällen ist oft sehr umfangreich und wenig änderungsfreundlich. Gleichzeitig ändern sich aber Anforderungen an die Software (und damit an den Test) sehr oft und oftmals auch noch kurz vor der geplanten Fertigstellung. Hr. Haller stand schließlich vor der Frage, seinem Team noch mehr Überstunden kurz vor der Freigabe der Magnetresonanz (MR)- Applikationen abzuverlangen oder sich etwas grundlegend Neues auszudenken.

Das brachte ihn und sein Team unter Mitwirkung der Firmen AFRA GmbH aus Erlangen und sepp.med gmbh aus Röttenbach im Frühjahr 2006 darauf, die Modellierung von Testszenarien in Enterprise Architect von SparxSystems in UML2 unter Zuhilfenahme des modellzentrierten Testens (.mzT) einzuführen und diese dann durch den Testfallgenerator MBTSuite automatisch in Testfälle umwandeln zu lassen.



Der Paradigmenwechsel

Es ist keine neue Erkenntnis, dass komplexe Sachverhalte durch Visualisierung übersichtlicher dargestellt und von Dritten leichter verstanden werden können. Darüber hinaus sind diese sehr viel einfacher zu überarbeiten bzw. zu ändern. Die Beschreibung eines Testszenarios eines „Systems under Test“ wird in geschriebener Prosa bei der gegebenen Komplexität schnell sehr lang und unübersichtlich. Durch Nutzung der Modellierung in der Modellierungsnotation UML können komplexe Zusammenhänge strukturiert dargestellt werden. Die sonst umständlich in Sprache zu fassenden Bedingungen und Voraussetzungen können in der visualisierten Logik, z. B. in beschreibenden Knoten und die Knoten verbindenden Kanten hinterlegt werden.

Diese Logik kann durch geeignete Algorithmen durchlaufen und abgeprüft werden, sodass daraus automatisch Testfälle erzeugt werden können. Aus der großen Anzahl von Werkzeugen, die eine Modellierung in UML2 erlauben, wurde „Enterprise Architect“ von SparxSystems ausgewählt. In Enterprise Architect können Requirements ebenso wie das dynamische Verhalten eines zu testenden Systems modelliert und verknüpft werden. Diese Modelle werden für die Generierung der Testfälle und Testdaten verwendet. Zum Einsatz kommt bei Siemens ein Algorithmus (siehe [OSS07] und [PSO08]), der in Zusammenarbeit mit der Universität Erlangen-Nürnberg als Forschungsprojekt der Hightech-Offensive Zukunft Bayern für den Einsatz zur Testfall- und Testdatengenerierung aus UML-Modellen entwickelt wurde. Dieser sogenannte „genetische“ Algorithmus basiert auf Erkenntnissen der biologischen Vererbungslehre, nach der sich für vorgegebene Umweltbedingungen besonders gut geeignete genetische Eigenschaften weiter vererben und durchsetzen. Die Übertragung dieses Mechanismus führt dazu, dass ausgehend von einem ersten Satz von Testfällen mit Testdaten weitere Inkremente erzeugt werden, die jeweils die „Gut“-Kriterien für Testfälle mit in die nächste Iteration weiter „vererben“ und damit die gewünschten Kriterien erreicht werden. Als Hauptkriterien dienen hier die Maximierung der Testabdeckung bei Minimierung des Testaufwands.

Mit dieser automatisierten Optimierungsstrategie hebt sich der „genetische Algorithmus“ von den bisher verwendeten Methoden in anderen am Markt erhältlichen Systemen zur Testgenerierung ab: In jenen muss der Anwender aus einer Auswahl möglicher Strategien (z.B. gewünschter Abdeckungsgrad aller Testfälle oder aber Auswahl von möglichen Pfaden über Knoten und Kanten) selbst über die Optimierungsmöglichkeiten entscheiden und diese ggfs. in mehreren Inkrementen vornehmen. Mit dem „genetischen Algorithmus“ wird dagegen sehr effektiv ein möglichst optimaler Satz an Testfällen erzeugt. Diese Testfälle können in beliebigen Formaten exportiert werden, sowohl in Textform für eine manuelle Durchführung als auch für die automatische Durchführung als Skript. Im konkreten Anwendungsfall werden die Testfälle direkt in das Testmanagement-Werkzeug „Quality Center“ von Hewlett Packard geschrieben, da dieses in dieser Abteilung bei Siemens verwendet wird. Es liegt auf der Hand, dass alle, die sich bislang mit dem Formulieren von Testfällen in einer fortlaufenden Prosa oder auch in Tabellen beschäftigt haben, umdenken und die Modellierung erlernen müssen. Um eine aktive Softwareentwicklung und die Freigabe gemäß einem verabschiedeten Projektplan nicht zu gefährden, wurde mit zunächst einem Teammitglied aus dem 13-Personen-Team, das sich mit dem Test eines eigenen abgegrenzten Teilprojekts beschäftigte, begonnen. Dieser Kollege wurde geschult und begann mit der Umsetzung.

In regelmäßigen Teambesprechungen wurde der Fortschritt berichtet und die gemachten Erfahrungen diskutiert. So konnten einmal gemachte Fehler für den Einsatz weiterer Kollegen in der Modellierung von vornherein vermieden werden.

Es empfiehlt sich, neben den bisherigen „normalen“ Teambesprechungen zum allgemeinen Projektfortschritt neue, spezielle Meetings für die Einführung und Umsetzung der neuen Methode durchzuführen. Dabei sollen „Lessons Learned“ ebenso klar angesprochen werden wie die Planung des weiteren „Roll-Outs“ sowie die Unterstützung durch die Leitung. Nur so kann vermieden werden, dass eine so grundsätzliche Neu-Einführung durch das Team schnell abgelehnt wird.

Am Ende der Einführung der Modellierung arbeiteten 5 der 13 Teammitglieder an der Modellierung und hatten erkannt und akzeptiert, dass für alle die neue Methode einen Vorteil bringt.

Das ist eine wesentliche Voraussetzung für die erfolgreiche Einführung: Alle müssen diese von sich aus einsetzen wollen, es muss ein „Selbstläufer“ werden. Wurden in einem ersten Projekt in der betrachteten Siemens-Abteilung zunächst 10% der zu testenden Anforderungen in 25 Modellen erstellt, so wurden diese für das dann erfolgreich durchgeführte Projekt auf 40% oder 29 Modelle ausgeweitet und damit 50% aller Testfälle (in Summe 151) automatisch erzeugt.

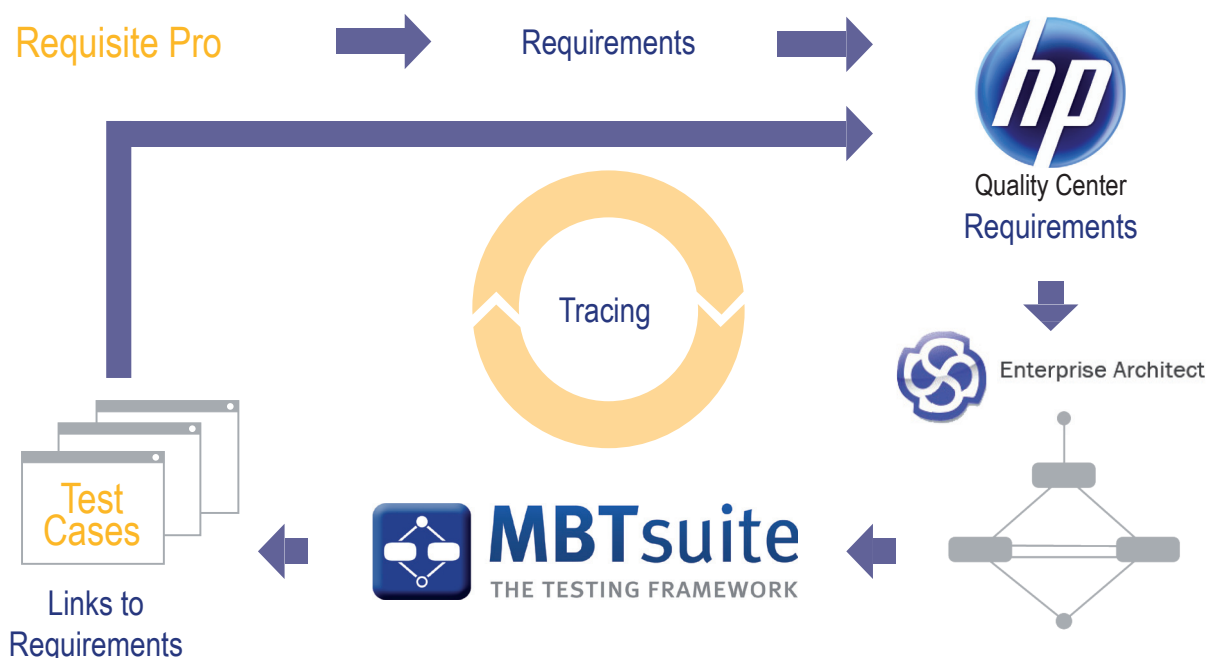
Änderungen im Prozess

Natürlich wird eine solche Neu-Einführung auch Auswirkungen auf vor- und nachgelagerte Dienststellen bzw. Abteilungen haben.

Bei Siemens Healthcare konnten die entsprechenden Prozesse mit nur minimalem Aufwand geändert werden. Man findet kaum noch Bereiche bei der Entwicklung und Herstellung von Medizintechnik-Systemen, die nicht schon durch moderne Software-Werkzeuge unterstützt werden, und Siemens ist sicher einer der Protagonisten darin. Daher war es eine entscheidende Voraussetzung für die erfolgreiche Implementierung der neuen Methode, diese und das dafür entwickelte Werkzeug vollständig in die bestehende Tool-Landschaft integrieren zu können. Die Werkzeuglandschaft beginnt mit Rational RequisitePro. Mit diesem Werkzeug werden die Anforderungen verwaltet und gepflegt.

- 1.) In Requisite Pro gesammelte und verwaltete Requirements werden automatisch mit dem HP Quality Center abgeglichen.
- 2.) Die Requirements werden für ein ausgewähltes Thema automatisch in das UML-Modell importiert und können dann vom Modellierer im Modell verlinkt werden.
- 3.) Die erstellten Modelle werden aus Enterprise Architect exportiert und von MBTsuite eingelesen. Daraus werden nach vorgegebenen Abdeckungskriterien automatisch Testfälle mit zugehörigen Testdaten erzeugt.
- 4.) Diese Testfälle werden ebenso automatisch ins HP Quality Center geschrieben und stehen mit Verlinkung zu den Anforderungen für die Testplanung und -durchführung zur Verfügung.
- 5.) Die Verbindung zwischen Requisite Pro und dem HP Quality Center macht es möglich, Testfälle und deren Durchführung mit den Testergebnissen den ursprünglichen Anforderungen zuzuordnen.

Dadurch entstand eine geschlossene Werkzeugkette:



Zwischenbilanz

Für den Test von syngo.via wurden 50% aller Testfälle mithilfe der neuen Methode .mzT und Nutzung des Werkzeuges MBTSuite erzeugt. Es konnte dieselbe Rate gefundener Fehler wie beim herkömmlichen Vorgehen ermittelt werden. Die Kosten dafür waren für den betrachteten Einführungszeitraum nahezu gleich. Damit ergibt sich jetzt, da sich der Prozess etabliert, die Erwartung, dass sich der Aufwand auf der Grundlage zunehmender Erfahrung signifikant reduzieren wird. Wichtige, bereits jetzt mehrfach gemachte positive Erfahrungen, sind klar erkennbar.

Die visuelle Darstellung des zu testenden Systems in einem übersichtlichen und strukturierten Modell ermöglicht es schon lange vor der Ausführung der Tests, Fehler in der Spezifikation zu erkennen und sich weitaus besser innerhalb eines weltweit kooperierenden Teams zu verständigen.

Die Kommunikation mit den Entwicklern wird einfacher, da sich beide Seiten auf der Ebene der logischen Abbildung schneller verstehen können, als dies durch eine unübersichtliche Prosa möglich wäre.

Die erzeugten Modelle sind besser anpassbar und dadurch nach einer Änderung schneller wieder zu verwenden. Reviews der erstellten Testszenarien müssen nicht durch ermüdendes Durchlesen dutzender Seiten oder unübersichtlicher Tabellen durchgeführt werden, sondern durch gemeinsames Durchgehen einer modellierten und auf einem Bild ersichtlichen Logik.

Eine geforderte Test-Abdeckung kann auf Modellebene durch den Einsatz der MBTSuite selbstverständlich nachgewiesen werden.

Ausblick

Der oben beschriebene Informationsaustausch über die Anwendung der neuen Methode in dafür speziell eingerichteten Besprechungen wird über die Einrichtung eines "Wiki's" institutionalisiert.

Tipps, Tricks, Erfahrungen und wichtige Neuigkeiten rund um das Thema Modellbasierter Test@MR können ins Intranet eingestellt und auch dort gelesen werden. In einem nächsten Schritt soll auch der Keyword driven Test über MBT ermöglicht werden.

Eine spannende Erweiterung ist durch die Anwendung des MBT in der agilen Entwicklung vorgesehen. Dabei können nach einem gemeinsamen Start von Entwicklung und Testmodellierung für einzelne Sprints letztere leicht für den nächsten Sprint angepasst und erweitert werden. Testfälle werden schnell für jedes Inkrement und jeden Sprint erzeugt.

Im Projekt wurde erfolgreich der „genetische Algorithmus“ eingesetzt, um 100% Abdeckung bei minimalem Testaufwand zu erreichen. Mit der aktuellsten Version der MBTSuite stehen viele weitere Strategien zur Testfallgenerierung zur Verfügung. Mit „Fullpath“ werden beispielsweise zunächst alle möglichen Testfälle erzeugt und können anschließend unter zu Hilfe nahme von Filtlermechanismen auf das optimale Maß reduziert werden. Es werden verschiedenste Möglichkeiten zur Verfügung gestellt, das optimale Test-Set für den jeweiligen Bedarf zu erstellen. Zusätzlich werden erforderliche Testdaten automatisch generiert und eine Requirements-Abdeckungsmatrix angeboten

Schlussfolgerung

Mit einer intelligenten und folglich reibungslosen Einführung des MBT können die Qualität der Ergebnisse gesteigert sowie der Einsatz von Ressourcen reduziert werden. Es gelingt somit eine gleichzeitige Steigerung der Effektivität und Effizienz.. Von entscheidender Bedeutung für die erfolgreiche Einführung ist die gleichzeitige konsequente Einbindung aller involvierten Mitarbeiter sowie die Gewährleistung eines permanenten Kommunikationsprozesses durch das Management. So wird die Anerkennung und Mitwirkung aller Beteiligten gesichert und die Methode und das Werkzeug werden zum „Selbstläufer“.

Referenzen:

[OSS07] Oster, N.; Schieber, C.; Saglietti, F.; Pinte, F.: Automatische, modellbasierte Testdatengenerierung durch Einsatz evolutionärer Verfahren. In Informatik 2007 – Informatik trifft Logistik, Lecture Notes in Informatics, Vol. 110, Gesellschaft für Informatik, 2007

[PSO08] Pinte, F.; Saglietti, F.; Oster, N.: Automatic Generation of Optimized Integration Test Data by Genetic Algorithms. In Software Engineering 2008 – Workshopband, Lecture Notes in Informatics, Vol. 122, Gesellschaft für Informatik, 2008



Der Autor Gerhard Baier hat Elektrotechnik an der TU Darmstadt studiert und ist seit 11 Jahren bei AFRA tätig. In dieser Zeit hat er beginnend mit der Entwicklung von Testsystemen für Embedded Systems über Testautomatisierung im medizintechnischen Bereich diverse Projekte durchgeführt und später dann erfolgreich geleitet. Inzwischen ist Herr Baier Mitglied der Geschäftsleitung der AFRA GmbH und leitet die Entwicklungsabteilung für Testsysteme und Prüfplätze. Auf Herstellerseite war er für die erfolgreiche Einführung des modellbasierten Tests im vorgestellten Projekt verantwortlich.



Florian Prester studierte an der Friedrich-Alexander Universität Erlangen-Nürnberg Informatik und war nach seinem Abschluss als wissenschaftlicher Mitarbeiter im Regionalen RechenZentrum der Universität tätig. Seit 2007 ist er bei der sepp.med gmbh beschäftigt. Er baute dort die firmeneigene F&E auf, mit Schwerpunkt Produktentwicklung der MBTsuite. Seit 2009 ist Florian Prester einer von drei Geschäftsführern der sepp.med gmbh wo er den Aufgabenbereichen F&E & Vertrieb – kurz: das operative Geschäft – verantwortet. Sein besonderes Augenmerk liegt auf der Entwicklung der Methodik modellzentriertes Testen (.mzT) und der Produktentwicklung des Testfallgenerators MBTsuite.

sepp.med
Qualität sichert Erfolg

sepp.med gmbh
Gewerbering 9
91341 Röttenbach

Fon: +49 (0) 91 95 - 9 31 - 0
Fax: +49 (0) 91 95 - 9 31 - 300

E-Mail: info@seppmed.de
Web: www.seppmed.de

AFRA GmbH
Gesellschaft für anwenderfreundliche Automatisierung

AFRA GmbH
Henkestraße 77
91052 Erlangen

Fon: +49 (0) 91 31 - 82611 - 0
Fax: +49 (0) 91 31 - 82611 - 40

E-Mail: info@afra.de
Web: www.afra.de
