

6. Ausblick und Zusammenfassung

You can only manage what
you can measure.

*(Peter Drucker,
US-amerikanischer Ökonom,
1909 †2005)

Inhalt

6.1. Erweiterungsmöglichkeiten	49
6.2. Zusammenfassung	52

Dieses Kapitel dient als Abschluss der Bachelorarbeit. Im folgenden Abschnitt werden mögliche Erweiterungen des Metrikvorschlagsystems diskutiert. Unter anderem werden sicherheitsrelevante, implementierungstechnische und fachliche Erweiterungen erläutert. Es folgte eine Zusammenfassung der Arbeit, die einen kurzen Überblick über die vorliegende Arbeit bietet.

6.1. Erweiterungsmöglichkeiten

Dieser Abschnitt beschäftigt sich mit sinnvollen Erweiterungen des Metrikvorschlagsystems. Dabei geht es sowohl um Implementierungsdetails als auch grundsätzliche Erweiterungen.

Als erste Erweiterung ist die erwähnte Administratoroberfläche zu nennen. Sinnvoll wäre hier das Löschen von Konten zu ermöglichen. Außerdem sollte es möglich sein, Kunden und Experten eines Metrikvorschlags auszutauschen, wenn diese beispielsweise kündigen. Es bietet sich auch an das Wechseln des Experten, was momentan die Experten selbst vornehmen können, einem dedizierten Administrator zu überlassen, um etwaige Manipulationen zu vermeiden. Momentan wäre es möglich, dass sich die Experten die Metrikvorschläge immer gegenseitig zuschieben. Dies würde mit einer Administratoroberfläche verhindert. Es muss evaluiert werden, ob dies ein Problem darstellt. Durch das Einschalten eines Administrators würden auch Wechsel der Zuständigkeit deutlich komplizierter ausfallen. Zusätzlich zum Expertenwechsel könnte der Administrator bei Bedarf Vorschläge löschen. Die Frage, die man sich diesbezüglich aber stellen muss, ist, wann dies sinnvoll ist. Im Allgemeinen sollten alle Metrikvorschläge zur Dokumentation und zur Sammlung des Wissens und der Erfahrung der Softwareorganisation gesammelt werden. Dem würde eine Löschung von Metrikvorschlägen

widersprechen. Allerdings könnte es sein, dass überflüssige Vorschläge gemacht werden, die keinen Nutzen besitzen und nur das Datenvolumen erhöhen. Außerdem könnte es passieren, dass aus Versehen ein nicht beabsichtigter, leerer Vorschlag eingereicht wurde, der wieder gelöscht werden soll. Das Löschen sollte aber offensichtlicherweise von Personen mit einer speziellen Administratorrolle ausgeführt werden, da ansonsten Experten unliebsame Vorschläge oder Bewertungen löschen könnten.

Eine weitere Verbesserung der Oberfläche wäre eine Suchfunktion. Beispielsweise könnte man nach der Vorschlagsnummer suchen. Sinnvoll wäre auch ein Intervall von Vorschlagsnummern angeben zu können, um beispielsweise ältere Vorschläge zu finden. Zusätzlich wäre eine Suche nach dem letzten Änderungsdatum sinnvoll, um Vorschläge aus einem bestimmten Zeitintervall zu finden. Zusätzlich könnte man eine Suche nach dem Metriknamen und dem Informationsbedarf anbieten. Allerdings muss man hier bei der Implementierung auf möglich Tippfehler der Nutzer achten. Letztendlich könnte die Erweiterung um eine Suchfunktion sicher am Schnellsten vorgenommen werden.

Aus Implementierungssicht wäre es unter Umständen sinnvoll für den Versand eine *Message-Driven Beans* zu nutzen. Momentan ist dies nicht notwendig, da der Empfängerkreis sehr gering ist. Maximal alle Experten bekommen gleichzeitig eine Mail und deren Anzahl ist beim Kooperationspartner momentan auf drei beschränkt. Sollte es allerdings gewünscht sein, dass auch Bilddateien in der E-Mail mitversandt werden, so dass die Datenmenge der E-Mails zunimmt, könnte das Versenden der E-Mails die aufrufenden Session Beans zu lange blockieren, da diese synchron arbeiten. Falls die E-Mails also an Datenvolumen gewinnen und so den Ablauf verzögern, wäre es sinnvoll das Versenden der E-Mails von einer Message-Driven Bean vornehmen zu lassen. Diese würde dann vom MetricProposalController eine Nachricht bekommen, dass sie ihre Arbeit aufnehmen kann, während der MetricProposalController nach Versand dieser Nachricht direkt weiterarbeiten kann. Das würde den Ablauf deutlich beschleunigen.

Weitere wichtige Erweiterungen betreffen die Zugriffsbeschränkungen des Webservices. Solange die Weboberfläche benutzt wird, sind keine illegalen Zugriffe auf das System möglich, da falsche Eingaben verhindert werden, da beispielsweise die Textfelder geblockt werden oder Buttons bzw. Links nicht angezeigt werden. Dies gilt natürlich nur, wenn noch ein Passwortschutz eingebaut wird, da momentan keine Passwortprüfung stattfindet. Da die Webservicemethoden nicht prüfen, wer sie aufruft, ist es möglich, dass beispielsweise jemand anderes als der jeweilige Kunde den Metrikvorschlag bewertet. Dies ist der Fall, weil dies nicht die fachliche Logik der Methoden betrifft. Aus Gründen der Kapselung sollten diese auch keinen technischen, sicherheitsrelevanten Code beinhalten. Um aber dennoch für Zugriffssicherheit zu sorgen, könnte *Interzeptoren* eingesetzt werden. Diese sind auch Teil der EJB 3.0-Spezifikation und ermöglichen eine *aspektorientierte Programmierung* (vgl [19]). Der Quelltext wird dabei nach Aspekten getrennt, so dass der fachliche Quelltext keinen sicherheitsrelevanten

Quelltext enthält. Dies verbessert die Verständlichkeit und Wartbarkeit des Codes deutlich, da nicht viele verschiedene Aspekte des Systems gleichzeitig überblickt werden müssen. Außerdem kann durch Trennung der Aspekte auch die Änderbarkeit verbessert werden, da bei Änderung der Sicherheitsregeln keine Änderung der Fachlogik nötig ist. Sollte beispielsweise der Einloggsvorgang geändert werden, ändert sich nicht das Einreichen neuer Metrikvorschläge. Die Interzeptoren können Methoden der EJBs umschließen, um so eine zusätzliche Funktionalität zu bieten. Wird nun die umschlossene Methode einer Bean aufgerufen, so werden zunächst die dazu annotierten Interzeptoren aufgerufen. Eine kleine Einführung in die Interzeptoren ist in Abschnitt [A.4.1](#) zu finden.

Im Zusammenhang mit der Zugriffssicherung sind außerdem *JAAS (Java Authentication and Authorization Service)* [27] und *LDAP (Lightweight Directory Access Protocol)* [25] zu nennen. JAAS dient dazu Rollen zu definieren, die bestimmte Methoden nutzen dürfen. Die Prüfung, ob der Aufrufende einer der erlaubten Rollen angehört, wird von JAAS nicht gewährleistet. Es werden lediglich Sicherheitsregeln und -einstellungen festgelegt. Die implementierten Sicherheitsregeln sind von der Technologie, die die Sicherheitsüberprüfung bereitstellt, unabhängig. Eine mögliche Technologie dafür wäre LDAP. Im LDAP-Directory-Server sind die verschiedenen Sicherheitsrollen mit Passwörtern definiert. Auf Seiten der EJBs wird ein *Sicherheitskontext* der Aufrufer definiert. Der Kontext wird durch den Anwendungsserver gestellt und der Kontext wird bei jedem Aufruf einer Enterprise Bean mitgeliefert. So kann in der Enterprise Bean geprüft werden, ob der Aufrufer die gewünschte Rolle innehat. Nutzt man also bestimmte Rollen bei den Methodenaufrufen, wie zum Beispiel einen Administrator, so kann man Benutzern den Zugriff auf diese Methoden verwehren. Dies entspricht genau dem gewünschten Verhalten für den Webservice. Der LDAP-Server würde die Zugriffsrechte beinhalten, sowie die Möglichkeit der Authentifizierung. Die EJBs nutzen Interzeptoren, damit der Sicherheitscode nicht in den Geschäftscode hineinspielt. Die Interzeptoren definieren Rollen, die für den Aufruf der Methoden, die sie umschließen, erlaubt sind. Sollte der Aufrufende nicht Mitglied der erlaubten Rollen sein, so wird ein Zugriff einfach abgelehnt. Wie man sieht ist diese Erweiterung sicherlich sehr wichtig, um das Metrikvorschlagssystem auch professionell einsetzen zu können. Allerdings ist diese Erweiterung auch sehr umfangreich.

Eine letzte Erweiterung betrifft die Integration in MeDIC. Ursprünglich war geplant das Metrikvorschlagssystem in den MeDIC-Editor zu integrieren. Die Metrikexperten hätten somit bei der Entwicklung und Pflege der Metriken die Kundenanfrage immer im Blick. Dieses Verschmelzen der beiden Projekte ist sicher der nächste Schritt um die Produktivität weiter zu steigern. Inzwischen ist aber das Ziel auch MeDIC auf EJB-Basis zu implementieren. Das verschmolzene System wird dann aber dennoch die sinnvolle Verknüpfung zwischen Metrikvorschlägen und tatsächlichen Metriken unterstützen. Erst dadurch wird sich der volle Nutzen der Systeme offenbaren.

6.2. Zusammenfassung

In dieser Bachelorarbeit wurde ein web-gestütztes, service-basiertes Ideenmanagementsystem zur Unterstützung bei der Definition von Metriken vorgestellt. Es dient der Ideensammlung von Metrikvorschlägen und der systematischen Unterstützung bei der Formulierung dieser.

Metriken sind eine bewährte Technik zur Qualitätssicherung und Projektplanung in Softwareorganisationen. Ihr Zweck ist die Vermessung von Produkten und Prozessen in einer Organisation, so dass diese „begreifbarer“ werden. Auf diese Weise wird die Identifizierung von Problemen oder die Verbesserung von Produkten und Prozessen ermöglicht. Allerdings ist ihre Verwendung in der Wirtschaft noch selten.

Bei der Entwicklung von Metriken werden Metrikprozesse angewendet, die unter anderem einen Ablauf für die Erstellung von Metriken vorgeben. Einer der vielfach eingesetzten Prozesse ist der *Goal Question Metric*-Ansatz. Er basiert auf einem Top-Down-Verfahren, bei dem zunächst das Ziel der Metrik definiert wird. Dann werden Fragen gestellt, die das Ziel zerlegen und so das Abstraktionsniveau senken. Aus den Fragen werden schließlich Metriken abgeleitet. Der beim Kooperationspartner eingesetzte *Goal Attribute Metric*-Prozess funktioniert ähnlich. Allerdings werden keine Fragen, sondern die Attribute des Messobjekts, zur Zerlegung genutzt.

Das entwickelte System wird sich voraussichtlich in MeDIC, ein vom *Lehr- und Forschungsgebiet Informatik 3 Softwarekonstruktion* entwickeltes Werkzeug zur Pflege, Verwaltung und Entwicklung von Metriken, integriert. MeDIC wird erst nach der Verwendung des hier entwickelten System eingesetzt, da für die Entwicklung einer Metrik im Allgemeinen zunächst ein Metrikvorschlag vorliegen muss.

Die Erstellung von Metrikvorschlägen erfolgt nach einem Metrikvorschlagsprozess. Während der aktuelle eingesetzte Prozess nicht ausreichend präzise definiert ist, ist der Prozess, dem das entwickelte System folgt, der Evaluation zufolge, ein wenig zu starr. Eine Anpassung an den vom Kooperationspartner geplanten Prozess wird in Zukunft erfolgen.

Das System besteht aus den zwei Hauptkomponenten *Serviceanbieter* und *Servicekonsument*. Der Serviceanbieter stellt einen Webservice zur Verfügung, den der Servicekonsument zur Darstellung mittels einer Weboberfläche nutzt. Die Funktionalität des Serviceanbieters wird durch Enterprise Java Beans 3.0 realisiert. Die während des Prozesses erzeugten Daten werden über JPA persistiert. Außerdem werden die Nutzer des Services per E-Mail über den neuesten Stand des Prozesses informiert.

Der Servicekonsument besteht aus Java Server Pages, die die eigentliche Vi-

sualisierung darstellen und Servlets, die die GUI-seitige Controllerfunktionalität realisieren. Zusammen verwirklichen sie eine übersichtliche Weboberfläche, die die Nutzer des Systems benutzen können, um mit dem Service zu interagieren.

Während der Entwicklung haben sich aufgrund der verwendeten Technologie einige Hindernisse ergeben, die in dieser Arbeit auch diskutiert wurden. Insbesondere die geforderte Festlegung auf WebSphere 6.1 als Anwendungsserver erwies sich als Problem.

Die vorgenommene Anwendungsstudie lieferte vielversprechende Ergebnisse, da die systematische Unterstützung als sehr hilfreich bewertet wurde. Auch die stetige Versorgung mit Informationen über den Fortschritt per E-Mail wurde positiv hervorgehoben. Die Datenbankunterstützung wurde als sehr sinnvoll bezeichnet. Allerdings ergaben sich auch Verbesserungsvorschläge, wie z.B. die farbkodierte Anzeige von Änderungen an den Metrikvorschlägen. Außerdem wurde um die Anpassung an den geplanten Prozess gebeten. Die Bearbeitung der Daten solle freier gestaltet werden, so dass keine Zustände eines Metrikvorschlags mehr existieren, die eine weitere Bearbeitung verhindern.

Es wurden jedoch nicht nur Erweiterungsmöglichkeiten vorgestellt, die sich in der Anwendungsstudie ergaben, sondern auch selbst erkannte Erweiterungen vorgestellt. Die Haupterweiterung betrifft dabei die Zugriffskontrolle auf die Methoden des Webservices, da aktuell jeder Benutzer die Methoden aufrufen kann. Außerdem wurde die Einführung einer Suchfunktion und einer Administratoroberfläche diskutiert.

