

Whitepaper

Medizinprodukte mit Risikoanalyse nach ISO 14971



Medizinprodukte mit Risikoanalyse nach ISO 14971

In unserer zunehmend vernetzten Welt steigt auch die durch Software realisierte Funktionalität in medizinischen Geräten. Mittlerweile gibt es neben den klassischen Standalone-Medizingeräten, die aus einer Steuereinheit und einer Benutzeroberfläche bestehen, sogar mobile Apps für Smartphones, die aufgrund ihrer medizinische Zweckbestimmung ebenfalls als Medizinprodukt zu bewerten sind.

Die notwendigen Schritte sind in den Normen IEC 60601, IEC 62304 und ISO 14971 fixiert und tragen dazu bei, die weiterhin zunehmende Interoperabilität zwischen verschiedenen Geräten und die steigende Komplexität der Medizinprodukte zu kontrollieren.

1 Überblick und thematische Einordnung

Das Medizinproduktegesetz (MPG) definiert, wann Software als Medizinprodukt bezeichnet werden kann, z. B. wenn der Hersteller eine medizinische Zweckbestimmung vorgibt. Dann ist Software ein integraler Bestandteil und kontrolliert u. a. die Geräteparameter.

Für die Entwicklung und das Inverkehrbringen von medizintechnischer Software fordert das MPG einen Softwarelebenszyklus (IEC 62304) sowie ein Risikomanagement (ISO 14971). Abbildung 1 umreißt die gesetzlichen und normativen Rahmenbedingungen. Abbildung 2 zeigt den Softwarelebenszyklus nach IEC 62304, der neben dem eigentlichen Softwareentwicklungsprozess auch die in der Norm definierten entwicklungsbegleitenden Prozesse enthält: Software-Risikomanagement, Software-Konfigurationsmanagement und Softwareproblemlösung.

AUTOR

Birgit Stehlik
infoteam Software AG

ZUSAMMENFASSUNG

- Risikomanagement nach ISO 14971
- Medizinische Softwareentwicklung
- Umsetzung normativer Anforderungen

KEYWORDS

ISO 14971, Risikoanalyse, Medizingeräte, Medical Devices, Mobile Apps

Version 1

WP-13-03-1

© 2013,
infoteam Software AG

2 Normenkonformer Softwarelebenszyklus

Eine wesentliche Anforderung aus der IEC 62304 ist die Nachverfolgbarkeit von Systemanforderungen, Softwareanforderungen, Softwaresystemtests und in der Software implementierten risikoreduzierenden Maßnahmen. Dies gilt für alle Software-Medizinprodukte, unabhängig von ihrer Klassifizierung. Daraus folgt, dass in die vollständige Abbildung eines Entwicklungsprozesses für medizinische Software auch der Risikomanagementprozess einfließen muss.

Damit diese Prozesse nicht nur theoretischen Charakter haben, sollte der Softwareentwickler auf Werkzeuge zurückgreifen, die ihn in seiner täglichen Arbeit darin unterstützen, die Vorgaben der IEC 62304 im Hinblick auf den Entwicklungsprozess einzuhalten.

infoteam hat aus diesem Grund einen Prozessleitfaden (iMED) realisiert, der die nach den Vorgaben der IEC 62304 und der ISO 14971 notwendigen Arbeitsschritte und normativen Anforderungen der einzelnen Projektphasen beschreibt. Dieser Prozessleitfaden ist unabhängig vom eingesetzten Tool und kann entsprechend adaptiert werden.

Für Softwareprojekte, die auf Microsoft® Windows-Betriebssystemen laufen und mit den entsprechenden Microsoft®-Entwicklungswerkzeugen und -Technologien entwickelt werden setzt infoteam den Microsoft® Team Foundation Server (TFS) ein. Für den TFS bilden wir den Entwicklungsprozess mit einem eigenen TFS-Prozess-Template und den im Prozessleitfaden definierten Workitems ab. Damit werden die Vorgaben der Normen IEC 62304 und ISO 14971 umgesetzt. Diese Umsetzung der normativen Anforderungen mit dem TFS dient im Folgenden als Beispiel zur Adaption des Entwicklungsprozesses nach dem Prozessleitfaden iMED. Selbstverständlich sind hierfür auch andere Tools oder Tool-Ketten denkbar.

Generell hängt die Wahl der verwendeten Tool-Kette von unterschiedlichen Rahmenbedingungen ab: Zum einen sind die Softwareentwicklungstools und eingesetzten Technologien zu betrachten, zum anderen die Schnittstellen zu Kundensystemen und Hardwareentwicklern sowie etablierte Infrastrukturen.

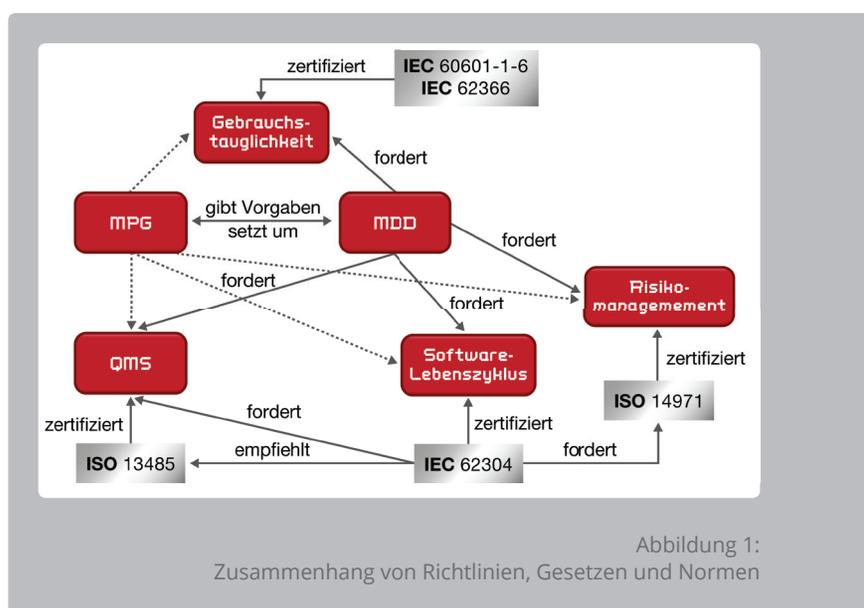


Abbildung 1: Zusammenhang von Richtlinien, Gesetzen und Normen

Der TFS stellt eine Sammlung von Tools zur Verfügung, die es erlauben, die einzelnen Tätigkeiten und Ergebnisse zentral zu verfolgen und zu koordinieren, so wie sie im Verlauf eines Projektes geplant und erarbeitet werden. Für die Entwicklung medizinischer Software heißt das, dass nicht nur ein gemeinsames Arbeitsverzeichnis für alle Entwickler existiert, sondern vielmehr eine komplette Softwareproduktionsumgebung: Quellverwaltung, Versionsverwaltung und Testautomatisierung sowie eine Kooperationsumgebung mit Dokumentenmanagement, Kommunikationsplattform und Prozessleitfaden sind hier integriert verfügbar. In einer gemeinsamen Datenbasis werden dabei auch die nicht Software-spezifischen Aufgaben (Tasks) verwaltet, die der Koordination an den Schnittstellen zwischen Software, Hardware und Mechanik dienen.

Vor allem in Projekten mit mehreren verteilt arbeitenden Projektpartnern und in großen Teams kommt der Vorteil eines solchen Systems zum Tragen. Denn der Zugriff auf die Projektdatenbasis erfolgt über das Netzwerk und über Web-Clients, was ein weltweit verteiltes Zusammenarbeiten ermöglicht.



Abbildung 2:
Softwarelebenszyklus nach IEC 62304

3 Rückverfolgbarkeit (Traceability)

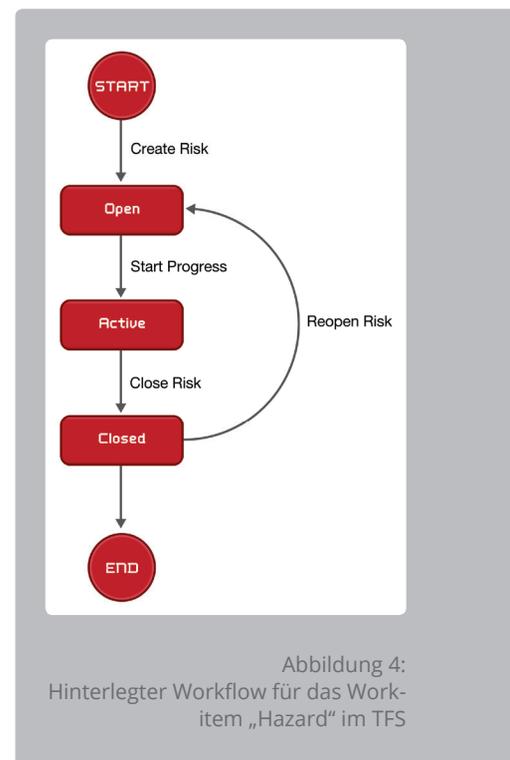
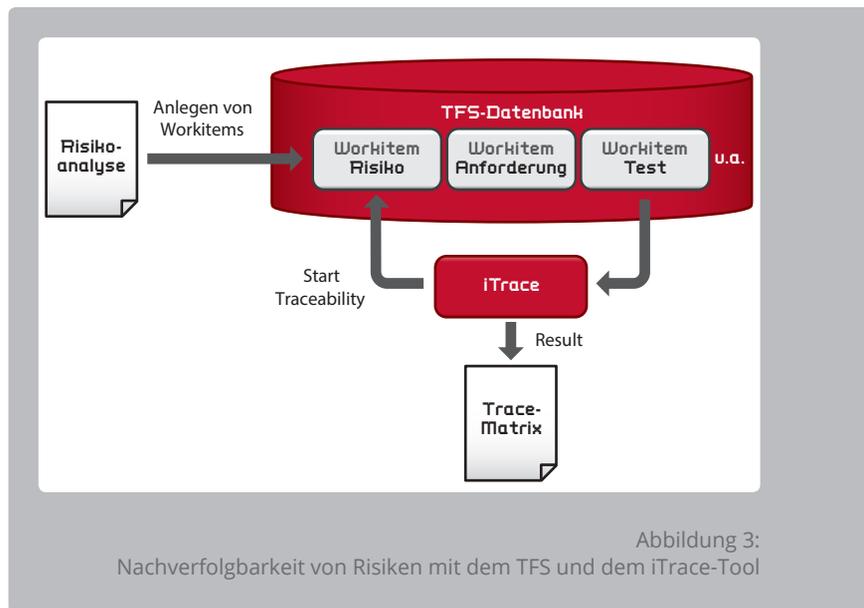
Die Anforderungen aus Spezifikations- und Architekturdokumentationen, die von den Entwicklern verfasst werden, können mithilfe eines Tools mit den Anforderungselementen (Workitems) im TFS synchronisiert und dort verwaltet werden. Ebenso verhält es sich mit erkannten Risiken aus der Risikoanalyse, die gleichfalls dokumentiert und referenziert werden.

Sofern ein Risiko durch eine in die Software zu integrierende Maßnahme reduziert werden kann, wird zu diesem Risiko ein Workitem „Anforderung“ angelegt, das mit einem Workitem „Test“ verknüpft ist. Damit wird sichergestellt, dass einerseits alle notwendigen Maßnahmen zur Risikoanalyse in die Softwareentwicklung einfließen und andererseits diese implementierten Maßnahmen auch getestet werden. Aus einem Workitem „Anforderung“ lassen sich im TFS entsprechende Arbeitspakete (Tasks) ableiten, die einzelnen Entwicklern eines Projekts zugewiesen werden können.

Abbildung 3 zeigt, wie dieser Prozess mithilfe des TFS abgebildet werden kann.

Das Workitem „Hazard“ stellt die Verknüpfung zwischen den Risiken aus der klassischen Failure Mode and Effects Analysis (FMEA) und den Anforderungen an die Software her.

Jedem Workitem ist ein Workflow zugewiesen, der unterschiedliche Zustände und Zustandsübergänge definiert. Jedem einzelnen dieser Zustände sind Pflichtfelder zugeordnet, die vom Bearbeiter vor dem Wechsel in den nächsten Zustand ausgefüllt oder gesetzt werden müssen. Beispielweise ist beim Anlegen eines Workitems eine Beschreibung anzugeben und beim Übergang von Open zu Active muss eine Bewertung des Schweregrades erfolgen. Den vollständigen Workflow für das Workitem Hazard zeigt Abbildung 4.



4 Projektarbeit im regulierten Umfeld

Task ID	Task Name	Kind	Milestones	Milestones	Milestones	Milestones	TestCases	TestResults
11	Valid: Device is not set to a secure state in case of a software error	PROJECT/Product	None	0	0	1	1	0
20	Valid: Editor Overview	PROJECT/Instrumenting/ABC Editor	SRS	1	0	0	0	0
21	Valid: Step Editor	PROJECT/Instrumenting/ABC Editor	SRS	1	0	0	0	0
22	Valid: Configuration Editor	PROJECT/Instrumenting/ABC Editor	Protobuf	1	0	0	0	0
30	Valid: Installation version management	PROJECT/Instrumentation/ABC Editor	SRS	1	0	0	0	0
31	Valid: Access Rights	PROJECT/Product	SRS	1	0	0	1	0
32	Valid: User Management	PROJECT/Product	SRS	1	0	0	1	0
120	Valid: Knowledge tool (Status)	PROJECT/Product	ChangeRequest	0	0	0	0	0
121	Valid: Knowledge tool (Status)	PROJECT/Product/Visualization	System/Instrument	0	0	0	0	0
122	Valid: Knowledge tool (Status)	PROJECT/Product	System/Instrument	0	0	0	0	0
123	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
124	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
125	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
126	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
127	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
128	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
129	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
130	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
131	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
132	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
133	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
134	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
135	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
136	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
137	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
138	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
139	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
140	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
141	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
142	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
143	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
144	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
145	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
146	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
147	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
148	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
149	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
150	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
151	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
152	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
153	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
154	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
155	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
156	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
157	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
158	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
159	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
160	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
161	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
162	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
163	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
164	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
165	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
166	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
167	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
168	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
169	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
170	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
171	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
172	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
173	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
174	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
175	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
176	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
177	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
178	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
179	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
180	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
181	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
182	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
183	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
184	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
185	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
186	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
187	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
188	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
189	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
190	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
191	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
192	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
193	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
194	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
195	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
196	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
197	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
198	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
199	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0
200	Valid: Knowledge tool (Status)	PROJECT/Product/Physical Definition	System/Instrument	0	0	0	0	0

Abbildung 5: Trace-Matrix mit Querverweisen zu den Dokumenten, verlinkten Tests und vorhandenen Testergebnissen

Da die IEC 62304 das Vorgehensmodell im Entwicklungsprozess nicht vorschreibt werden bei der Entwicklung medizinischer Software sowohl iterativ-inkrementelle als auch agile Ansätze verfolgt.

Wichtig für das Risiko- und Projektmanagement sind Tools und Datenbankabfragen, die das Projektcontrolling unterstützen. Damit lassen sich beispielsweise Informationen über die Anzahl der gemeldeten oder gelösten Fehler gewinnen oder eine Prüfung der Nachverfolgbarkeit durchführen. Hierbei werden die aktuellen Bearbeitungszustände der Abhängigkeiten eines Hazards abgefragt und angezeigt, z. B. zu welchen Hazards eine Anforderung vorhanden ist.

Bevor nun ein Produkt vom Kunden abgenommen werden kann, stellen sich Entwickler stets dieselben Fragen: Sind alle Anforderungen getestet? Waren diese Tests erfolgreich oder schlugen einzelne fehl? Gibt es Softwarekomponenten, für die noch gar keine Tests definiert wurden? Um diese Fragen gewissenhaft und nachverfolgbar beantworten zu können, bewertet das von infoteam entwickelte Tool iTrace sämtliche Testergebnisse und stellt sie übersichtlich in Form einer Trace-Matrix dar (siehe Abb. 5).

5 Zusammenfassung

Konkrete Erfahrungen mit dem skalierbaren und zukunftsorientiert ausgerichteten Projektmanagement für die Entwicklung medizinischer Software zeigen, dass besser geführte und transparent dargestellte Prozesse die Effizienz wesentlich steigern und auch den Nachweis gegenüber benannten Stellen deutlich erleichtern.

6 Glossar

FMEA	Failure Mode and Effects Analysis
IEC	International Electrotechnical Commission
ISO	International Standardization Organization
MPG	Medizinproduktegesetz
TFS	Microsoft® Team Foundation Server

Kontakt

infoteam Software AG

Am Bauhof 9
D-91088 Bubenreuth

Telefon: +49 (0) 9131 / 78 00 - 0
Telefax: +49 (0) 9131 / 78 00 - 50
info@infoteam.de
www.infoteam.de



infoteam Software AG

Emil-Figge-Straße 80
D-44227 Dortmund

Telefon: +49 (0) 231 / 97 42 56 - 00
Telefax: +49 (0) 231 / 97 42 56 - 09
dortmund@infoteam.de
www.infoteam.de



infoteam Software AG

Laubisrütistrasse 44
CH-8712 Stäfa

Telefon: +44 (0) 44 927 15 15
Telefax: +44 (0) 44 927 15 10
info@infoteam-software.ch
www.infoteam-software.ch



infoteam Software (Beijing) Co., Ltd.

Zhongguancun North Street 151
Yan Yuan Resource Tower, Room 820
100080, Haidian District Beijing China

Telefon: +86 (0) 10 5887 6786
Telefax: +86 (0) 10 5887 6785
info@infoteam.com.cn
www.infoteam.com.cn

