



Marc Bless

Scrum und die IEC 62304

**Medizinische Software mit agilen Methoden
normkonform entwickeln**

Vollständige Ausgabe

Version 1.2

Juli 2013, Baden-Baden

Schlagworte:

Medizinische Software, IEC 62304, agile Methoden, Scrum, Medizintechnik, Normkonformität, Software-Lebenszyklus, Software-Life-Cycle, Softwareentwicklung

© Marc Bless, alle Rechte vorbehalten

www.agilecoach.de

Version 1.2

Gestaltung: Marc Bless

Baden-Baden, Juli 2013

Alle Rechte vorbehalten. Insbesondere das Recht der mechanischen, fotografischen oder elektronischen Vervielfältigung, der Einspeicherung und Verarbeitung in elektronischen Systemen, des Nachdruck in Zeitschriften und Zeitungen, des öffentlichen Vortrages, der Verfilmung oder Dramatisierung, der Übertragung durch Rundfunk, Fernsehen und Video, auch einzelner Bild- oder Textteile sowie der Übersetzung in andere Sprachen.

Inhaltsverzeichnis

Vorwort von Frank Lange	7
Über dieses Buch	9
Warum Agil und Medizintechnik?	11
Schneller Markteinstieg	11
Kundenzufriedenheit und Qualität	11
Agile Methoden - ein Mysterium?	13
Verbindung zweier Welten	14
Von der Norm zum Prozess	16
Überblick der Prozesse und Artefakte	23
Prozesse, Meetings, Reviews	28
Acceptance Testing	29
Clean Code / SOLID Principles	33
Continuous Integration	36
Early Integration / Early Testing	38
Exploratory and Manual Testing	40
Integration Testing	42
Problem Communication	45
Product Backlog Grooming	47
Quality Management Prozess	53
Release Planning Meeting	55
Risk Analysis and Management	58
Scrum	65

Software Maintenance Process	67
Sprint Planning	69
Sprint Retrospective	75
Sprint Review	77
Team Kick-Off Meeting	85
Test Automation	87
Test-Driven Development (TDD)	92
Unit Testing	97
Usability Process	100
Zero Bug Tolerance	102
Dokumente, Artefakte, Pläne	106
Architecture Documentation	107
Definition of Done	109
Definition of Ready	113
Iteration Notes (Sprint Notizen)	116
Linkable IDs	118
Product Backlog	120
Release Notes	123
Software Development Plan	125
Software Maintenance Plan	128
Sprint Development Plan = Sprint Backlog	130
Team Charter / Working Agreements	133
Test Documentation	136
User Story	138
Versioning System	145

Von den Anforderungen der Norm zu Praktiken und Artefakten **148**

§ 1 - Scope	149
§ 4 - General requirements	151
§ 5 - Software development process	155
§ 5.1 - Software development planning	155
§ 5.2 - Software requirements analysis	162
§ 5.3 - Software architectural design	169
§ 5.4 - Software detailed design	172
§ 5.5 - Software unit implementation and verification	175
§ 5.6 - Software integration and integration testing	179
§ 5.7 - Software system testing	184
§ 5.8 - Software release	190
§ 6 - Software maintenance process	193
§ 6.1 - Establish software maintenance plan	193
§ 6.2 - Problem and modification analysis	196
§ 6.3 - Modification implementation	199
§ 7 - Software risk management process	201
§ 7.1 - Analysis of software contributing to hazardous situations	201
§ 7.2 - Risk control measures	204
§ 7.3 - Verification of risk control measures	206
§ 7.4 - Risk management of software changes	208
§ 8 - Software configuration management process	209
§ 8.1 - Configuration identification	209
§ 8.2 - Change control	211
§ 8.3 - Configuration status accounting	213

§ 9 - Software problem resolution process	214
Literatur- und Quellenverzeichnis	218
Feedback und Kontakt	223

Vorwort von Frank Lange

Wenn ich bei Vorträgen oder in Gesprächen mit Kunden auf das Thema Scrum in der Medizintechnik zu sprechen komme, blicke ich oft in erstaunte Gesichter. "Dürfen wir das überhaupt?" ist meist die erste Frage, noch bevor der Nutzen geprüft wird. Klar, Scrum und andere agile Methoden sind mittlerweile in aller Munde und haben in anderen Branchen bereits zu vielen erfolgreichen Projekten geführt - aber in der Medizintechnik? Das geht dann vielen doch ein wenig zu weit.

Produktentwicklungen in der Medizintechnik sind gekennzeichnet von einem hohen Qualitätsbewusstsein. Es geht um Menschenleben, und jedes potentielle Produktisiko muss daher eliminiert oder zumindest auf ein Minimum reduziert werden. Normen und Richtlinien sind notwendig um die erforderlichen Prozesse einzufordern, mit deren Hilfe diese hohe Produktqualität sichergestellt werden kann. Als Prozess hat sich in der Medizintechnik das V-Modell etabliert und in den letzten Jahrzehnten dafür gesorgt, dass Medizinprodukte immer sicherer, zuverlässiger und kostengünstiger entwickelt und hergestellt werden können.

Doch seit ein paar Jahren ändert sich der Markt für Medizintechnik rasant. Unternehmen müssen sich immer stärker spezialisieren, um Spitzenpositionen in einem bestimmten Marktsegment - und sei es auch nur eine kleine Nische - zu erlangen. Je enger die Nische, umso wichtiger ist ein weltweiter Vertrieb, denn nur so ist der Gesamtmarkt dieser Nische groß genug, um die immer größer werdenden Entwicklungskosten zu decken. Diese Anforderung zur Spezialisierung und Internationalisierung gilt auch - und speziell für kleinere - Unternehmen. Als dritte Anforderung kommt dazu, dass Entwicklungen in der Medizintechnik immer schnelleren Zyklen unterliegen. Ursache hierfür ist, dass Kunden Medizinprodukte nicht mehr nur mit dem direkten Wettbewerb vergleichen, sondern immer häufiger auch mit Consumerprodukten wie Smartphones, Tab-

lets u.ä. Von Medizintechnikprodukten wird die gleiche einfache Handhabung erwartet, denn Einarbeitungszeiten kann sich im Gesundheitssystem keiner mehr leisten. Zeit ist Geld!

Agile Methoden, speziell Scrum, bieten erprobte Lösungen für diese neuen Anforderungen. Scrum ist seit weit über einem Jahrzehnt etabliert, viele durchgeführt Projekte belegen den Erfolg. Aber Scrum ist so... anders! Ist man es bisher in der Medizintechnik eher gewohnt, hunderte von Seiten in den Normen durchzuarbeiten um klare Arbeitsanweisungen daraus abzuleiten, bauen agile Methoden stärker auf Werten auf. Das agile Manifest besteht im wesentlichen aus nur vier Zeilen - kann das funktionieren? Und steht nicht im agilen Manifest gar etwas davon, dass nicht mehr dokumentiert werden muss? Das wäre ja ein direkter Widerspruch zur Norm! Solche scheinbaren Widersprüche können dazu führen, dass man sich lieber gar nicht erst mit Scrum auseinandersetzt, um auf der sicheren Seite zu bleiben. Auch wenn diese "sichere Seite" alles andere als sicher ist - siehe oben.

Marc Bless hat sich mit diesen Widersprüchen auseinandergesetzt. Intensiv. Er arbeitet als agiler Coach, Trainer und Berater und hat daher Scrum "im Blut". Gleichzeitig hat er jahrelange Erfahrungen im Bereich der Medizintechnik gesammelt und kennt sowohl die Normen und Anforderungen als auch das "Mindset" der Branche. Ihm geht es nicht um ein Entweder-Oder zwischen Normen und Scrum sondern um ein Sowohl-als-auch. Und Marc ist Niemand, der sich mit Kompromissen zufrieden gibt. Was Sie in diesem Buch lesen, hat den Praxistest bereits hinter sich. Es ist die Essenz aus vielen Jahren Erfahrung mit Scrum speziell in der Medizintechnik. Und es funktioniert.

Ich wünsche Ihnen viele gute neue Gedanken bei der Lektüre dieses Buches und eine erfolgreiche Umsetzung in die Praxis!

Frank Lange
Pfinztal, den 08. Juli 2013

Über dieses Buch

Sie als Leser haben in diesem Augenblick eines von zwei möglichen Varianten dieses Buches in der Hand¹ und ich möchte Ihnen kurz erklären, was Ihre Variante von der existierenden zweiten unterscheidet.

Vollständige Ausgabe

Die vollständige Ausgabe beinhaltet neben dem Ergebnis meiner eigenen Arbeit auch einen Großteil der Originaltexte aus der Norm IEC 62304. Dadurch erhalten Sie die Möglichkeit, die vorgeschlagenen und empfohlenen Methoden, Praktiken und Artefakte direkt mit den Anforderungen aus der Norm abzugleichen. Diese Variante des Buches ermöglicht bei Anwendung agiler Methoden eine leichte Auditierbarkeit, da für sämtliche Anforderungen aus der Norm ein formales Abbild in Form von Praktiken und Artefakten beschrieben und schnell ersichtlich ist.

Durch die enthaltenen Originaltexte aus der Norm kommen für jedes verkaufte Exemplar Lizenzrechte und Lizenzkosten der IEC zur Geltung. Damit ist die vollständige Ausgabe die kostspieligere Variante.

Gekürzte Ausgabe

Die gekürzte Ausgabe enthält keine Originaltexte aus der Norm IEC 62304. Das Ergebnis meiner eigenen Arbeit referenziert lediglich auf einzelne Paragraphen der Norm. Diese Referenzierung ermöglicht immer noch die Auditierbarkeit der angewandten agilen Methoden. Der Nachweis erfordert jedoch

¹ oder als elektronische Version auf Ihrem Gerät.

auf Grund des fehlenden, formalen Abbildes eine umfangreiche Eigenleistung Ihrerseits.

Durch die fehlenden Originaltexte aus der Norm entstehen keine weiteren Lizenzkosten der IEC. Damit ist die gekürzte Ausgabe die günstigere Variante.

Warum Agil und Medizintechnik?

Schneller Markteinstieg

Medizintechnik und die Entwicklung entsprechender Medizinprodukte befinden sich in einem regulierten Umfeld. Gesetze, Normen und Richtlinien geben mehr oder weniger harte Rahmenbedingungen vor, in deren Grenzen ein Medizinprodukt umgesetzt werden darf. Die Einhaltung dieser Rahmenbedingungen führt im klassischen Projektmanagement oft zu langen Entwicklungszyklen, in denen wir erst sehr spät feststellen können, ob unsere anfänglichen Annahmen mit ihrer Realisierung auch tatsächlich marktreif und im Interesse der Anwender umgesetzt wurden.

Lange Entwicklungszeiten für Produktlebenszyklen sind wirtschaftlich heutzutage nicht mehr sinnvoll. Eine kurze Time-to-Market wird immer wichtiger, um den Marktbegleitern einen Schritt voraus zu sein und innovative Produktideen schnell als erster am Markt platzieren zu können. Und selbst Me-Too-Ansätze in der Produktentwicklung profitieren von einer schnellen Umsetzung der Ideen Anderer - schließlich wollen Sie Ihr Me-Too-Produkt anbieten können, lange bevor der Innovator Ihren Markt mit der nächsten, neuen umgesetzten Idee zu Ihren Ungunsten verändert.

Kundenzufriedenheit und Qualität

Der Kunde eines Medizinproduktes wird immer mündiger und entscheidet selbst über den Erfolg einzelner Produkte und

Dienstleistungen. Dies erfordert bereits bei der Entstehung neuer Produktideen eine gewisse Nähe zum Anwender oder sogar die entwicklungsbegleitende Einbindung echter Anwender. In traditionellen Vorgehensweisen kann der Anwendungserfolg oft erst ganz am Ende des Prozesses ermittelt werden. Ein entsprechendes Reagieren auf Kunden und Anwender² sowie deren wertvolles Feedback kann dann erst in einem nächsten Produkt, Projekt oder Release stattfinden und sich auswirken.

Der Kunde bewertet ein Produkt nicht nur nach seinen Funktionen und der einfachen Handhabung, sondern auch nach seiner Qualität und Fehlerfreiheit. Dabei spielen oft schon Kleinigkeiten eine große Rolle, die zu einer negativen Bewertung eines Produktes führen können. In sozialen Netzwerken und Produktbewertungsportalen multiplizieren sich diese Effekte mittlerweile soweit, dass wir feststellen müssen: über Sieg oder Niederlage eines Produktes kann heute die breite Masse der Anwender entscheiden, unabhängig von Marketing- und Werbemaßnahmen.

Dabei verlassen wir uns im Regelfall auf die Empfehlungen unserer (oftmals sehr weitläufigen) Freunde und Kontakte. Wir vertrauen mehr auf deren Urteilsvermögen und Erfahrungen als auf ausgeklügelte Werbebotschaften. Hier gilt es also, von Anfang an eine stets hohe Qualität sowie einen großen Anwendernutzen an den Markt zu bringen, um die Kundenzufriedenheit zu maximieren.

² Im weiteren Text wird nicht mehr strikt zwischen Kunde (derjenige, der das Produkt bezahlt) und Anwender (derjenige, der das Produkt tatsächlich benutzt) unterschieden, um die Lesbarkeit des Textes nicht unnötig zu strapazieren.

Agile Methoden - ein Mysterium?

Der Einsatz agiler Management- und Entwicklungsmethoden liegt auf der Hand. Schnelle Entwicklungszyklen, Einbindung und Feedback echter Anwender, sowie höchste Qualitätsansprüche können durch agile Methoden angeblich nachhaltig realisiert werden. Des Weiteren haben agile Methoden in den letzten Jahren ein breites Publikum erreicht und sind in aller Munde. Immer mehr Unternehmen widmen sich diesem Thema, also muss etwas dran sein. Oder handelt es sich bei Scrum und co. einfach nur um einen aktuellen Trend, dem nächstes Jahr wieder etwas anderes folgt?

Viele Versprechen, aber auch Missverständnisse kursieren bezüglich agiler Methoden:

- „Scrum ist chaotisch.“
- „Jeder macht, was er will.“
- „Es wird nichts mehr dokumentiert.“
- „Scrum funktioniert nur bei kleinen Web-Projekten.“
- „Agilität funktioniert nicht bei komplexen Medizintechnik-Projekten.“
- „Agile Methoden funktionieren in unserer Organisation nicht.“
- „Scrum ist die Lösung all unserer Probleme.“
- „Durch Agilität werden wir effizienter.“
- „Die IEC 62304 beschreibt einen konkreten Software-Life-Cycle-Prozess und muss eins-zu-eins umgesetzt werden.“
- „Wir müssen ein V-Modell einsetzen, um normkonform zu arbeiten.“

Die Kombination der agilen Welt und des regulierten Umfeldes der Medizinproduktentwicklung bereitet vielen Leuten Kopfzerbrechen und führt zu großer Unsicherheit. In dieser Unsicherheit lauert die Gefahr, agile Methoden in bestehende, normkonforme Prozesse zu pressen, ohne dass sich der Vorteil echter Agilität entfalten kann.

Eine vereinfachte, beschränkte oder missverstandene Anwendung agiler Methoden kann dazu führen, dass das Scheitern eines Produktes oder Projektes eben diesen agilen Methoden zugerechnet wird:

- „Agil haben wir schon probiert, das hat ja auch nicht funktioniert.“
- „Agil hat bei uns alles nur schlimmer gemacht.“
- „Scrum hat uns mit Problemen konfrontiert, die wir vorher gar nicht hatten.“

Die Einführung von Scrum erfordert sehr viel Verständnis der zugrundeliegenden agilen Prinzipien, um nicht in die Falle zu laufen, fundamentale Fehler zu begehen. Diese Fehler mögen für den Unerfahrenen wie unwichtige Kleinigkeiten aussehen und oftmals ist es einfach bequemer, eine bestimmte Praktik oder Methode nicht einzuführen, da sie das bestehende Gefüge und die Unternehmenskultur im Innersten treffen können.

Wir³ wissen heute aus Erfahrung, dass agile Methoden wie Scrum funktionieren, wenn sie richtig und konsequent ein- und umgesetzt werden.

Verbindung zweier Welten

Können nun aber beide Welten effektiv nebeneinander existieren? Wie viel Agilität passt in die normativen Anforderungen? Welche agilen Praktiken eignen sich für einen regulierten Entwicklungsprozess?

³ Ich selbst aus jahrelanger Erfahrung als agiler Coach und viele meiner Kollegen, die sich intensiv damit beschäftigen, Unternehmen aller Branchen erfolgreich mit agilen Methoden und Prozessen zu transformieren.

Eine für uns grundlegende Aussage der Norm lautet, dass grundsätzlich jeder beliebige wasserfallartige, iterative oder evolutionäre Prozess verwendet werden kann⁴.

Die grundsätzliche Frage lautet: **wie kann die Normkonformität agiler Prozesse und Methoden nachgewiesen werden?**

Das vorliegende Buch gibt Antworten auf diese Frage und beschreibt eine vollständige, IEC 62304 konforme Implementierung von Scrum als Projektmanagementmethode, die mit agilen Entwicklungspraktiken aus dem Extreme Programming angereichert ist.

⁴ IEC 62304, Annex B

Von der Norm zum Prozess

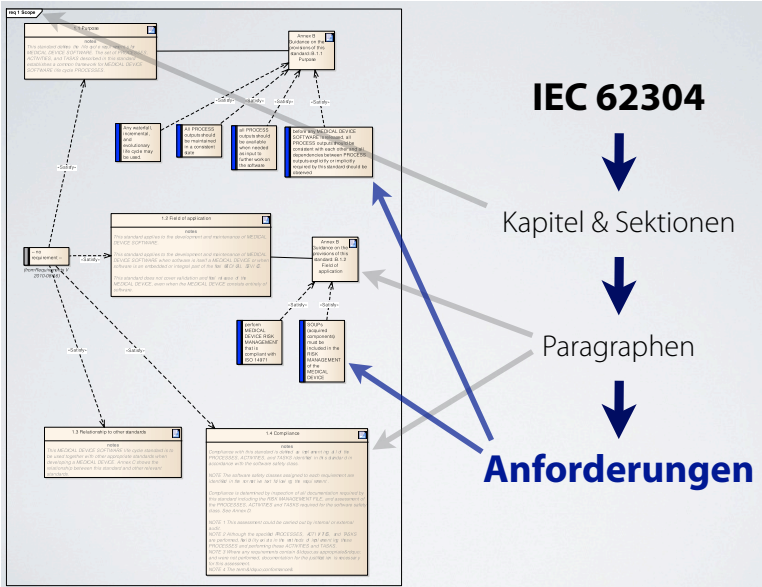
Der typische Weg für viele Qualitätsmanager und Prozessbeauftragte, eine Norm umzusetzen, ist die direkte Abbildung der Norminhalte auf einen Prozess. Damit erhalten wir im Regelfall einen vollständig normkonformen Prozess und dies ist in einer klassischen Organisation oft auch die Zielsetzung. Dort geht es nur in zweiter Linie um Effizienz der eingesetzten Methoden.

Die Abbildung der Norm auf einen eigenen, selbst definierten und normkonformen Prozess erfordert sehr viel mehr Aufwand, da dies eine detaillierte Auseinandersetzung mit der Norm voraussetzt und ein tiefes Verständnis des „Geistes“ einer Norm entwickelt werden muss. Mit „Geist“ ist die Meta-Ebene einer Norm gemeint, das heisst die darunter liegenden Prinzipien und die grundsätzliche Zielsetzung. Durch die Erarbeitung dieser Perspektive auf die normativen Inhalte wird es möglich, aus dem sequentiell beschriebenen Ablauf der Norm auszubrechen und zu einer neuen, teilweise kontra-intuitiven Umsetzung der Prozesse zu gelangen.

Um zu einem normkonformen, agilen Prozess zu gelangen, wurden die im Folgenden beschriebenen Transformationen der Norminhalte durchgeführt.

Anforderungen an den Software-Life-Cycle-Prozess

Zunächst wurde die Struktur der IEC 62304 vollständig aufgebrochen. Aus den einzelnen Kapiteln, Sektionen und jedem darin enthaltenen Paragraphen wurden die eigentlichen Anforderungen an einen Software-Life-Cycle-Prozess extrahiert.



Beispiel 1

Aus dem Paragraphen

§ 7.1.2 Identify potential causes of contribution to a hazardous situation

The MANUFACTURER shall identify potential causes of the SOFTWARE ITEM identified above contributing to a hazardous situation.

The MANUFACTURER shall consider potential causes including, as appropriate:

- a) incorrect or incomplete specification of functionality;*
- b) software defects in the identified SOFTWARE ITEM functionality;*
- c) failure or unexpected results from SOUP;*
- d) hardware failures or other software defects that could*

*result in unpredictable software operation; and
e) reasonably foreseeable misuse.*

[Class B, C]

wurden die folgenden, fünf Anforderungen an einen Software-Life-Cycle-Prozess extrahiert:

1. [B,C] consider potential causes including incorrect or incomplete specification of functionality
2. [B,C] consider potential causes including software defects in the identified SOFTWARE ITEM functionality
3. [B,C] consider potential causes including failure or unexpected results from SOUP
4. [B,C] consider potential causes including hardware failures or other software defects that could result in unpredictable software operation
5. [B,C] consider potential causes including reasonably foreseeable misuse

In diesem typischen Fall wurden die aufgelisteten Anforderungen aus dem Paragraphen in einzelne Anforderungen an den Software-Life-Cycle-Prozess extrahiert. Dies hat den Vorteil, dass für jede einzelne Anforderung die einfachste Möglichkeit gefunden werden kann, sie zu befriedigen.

Beispiel 2

Aus dem Paragraphen

§ 7.3.1 Verify RISK CONTROL measures

The implementation of each RISK CONTROL measure documented in 7.2 shall be VERIFIED, and this VERIFICATION shall be documented. [Class B, C]

wurden die folgenden beiden Anforderungen an einen Software-Life-Cycle-Prozess extrahiert:

1. [B, C] verify the implementation of each RISK CONTROL measure
2. [B, C] document the verification of implemented risk control measures

Auch hier wurden auf den ersten Blick zusammenhängende Fragmente aus dem Paragraphen aufgeteilt in einzelne Anforderungen. Die beiden Aspekte „Implementation“ und „Dokumentation“ können mit verschiedenen Praktiken umgesetzt werden, deswegen ist eine Trennung der beiden sinnvoll.

Vollständige Menge aller Anforderungen

Das Ergebnis dieser Transformation von Paragraphen in einzelne Anforderungen ist eine vollständige Menge aller dieser Anforderungen an den Software-Life-Cycle-Prozess.

Die Struktur der Norm suggeriert einen sequentiellen Entwicklungsablauf, der einem agilen Ansatz im Wesen widerspricht. Gleichzeitig stecken in dieser Struktur viele redundante Anforderungen, die zu einem sequentiellen Ablauf eher orthogonalen Charakter haben, wie zum Beispiel das gesamte Thema Risikomanagement.

Aus diesem Grund wurde die Menge aller Anforderungen gruppiert und in neue, einzelne Cluster eingeordnet, welche einzelne Entwicklungsthematiken inhaltlich abgeschlossen handhabbar machen.

Dokumente und Prozesse

Den Anforderungen in den Clustern wurden konkrete agile und klassische Praktiken und Artefakte zugeordnet. Die Zuordnungen einzelner Praktiken und Artefakte zu einzelnen Anforderungen wurden so vorgenommen, dass die Gesamtmen-

ge aller normativen Anforderungen vollständig erfüllt wird. Jede einzelne Zuordnung ist schriftlich begründet, um einen Nachweis der Anforderungserfüllung führen zu können.

Das Ergebnis aller Praktiken und Artefakte mitsamt ihren Begründungen findet sich im Hauptteil dieses Buches.

Das wichtigste Prinzip für die Zuordnung von Praktiken und Artefakten ist Einfachheit („Simplicity“) und bedeutet, die minimale, aber vollständige Befriedigung der normativen Anforderungen zu finden.

Praktiken sind dabei im Wesentlichen Prozesse, Reviews, Meetings, Besprechungen und Workshops.

Artefakte sind sämtliche Dokumente, Ergebnisse, funktionierende Software, Pläne und Beschreibungen, welche im Entwicklungsverlauf entstehen und eine Rolle spielen.

Nachweis der Normkonformität

Der Nachweis der Normkonformität wird für jede einzelne Praktik und jedes einzelne Artefakt in diesem Buch aufgeführt. Im Hauptteil des Buches wird dafür in jedem entsprechenden Unterkapitel auf die Paragraphen der Norm und deren im Einzelnen spezielle Anforderungen an den Software-Life-Cycle-Prozess verwiesen, sowie die Begründung aufgeführt, wie die Normkonformität zu jeder Anforderung hergestellt wird.

Der nachgelagerte Teil des Buches führt in tabellarischer Form durch die einzelnen Sektionen und Paragraphen der Norm und Listet die jeweiligen Praktiken/Artefakte auf mitsamt der Begründung der Normkonformität.

Dadurch wird für Anwender dieser Methodik sehr schnell ersichtlich, welche Praktiken und Artefakte für welche Paragraphen und Anforderungen aus der Norm eingesetzt werden können. Auditoren wird sehr leicht aufgezeigt, welche Para-

graphen der Norm mit welchen Praktiken und Artefakten umgesetzt werden können.

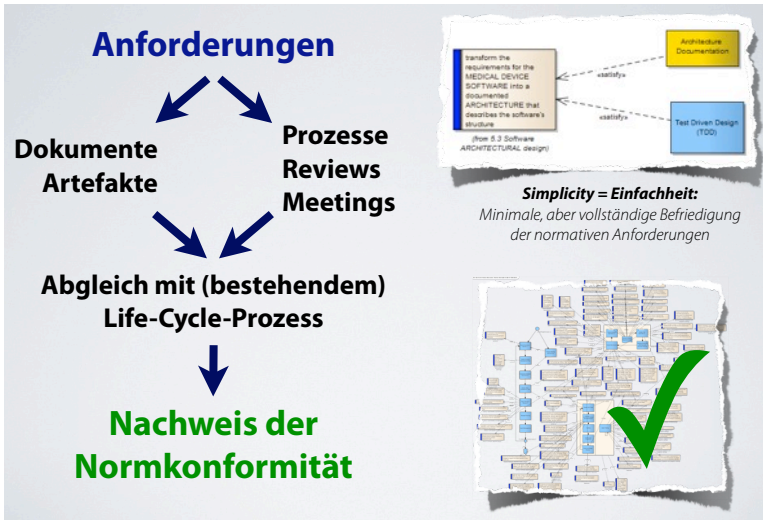
Angepasste Vorgehensmodelle

Das in diesem Buch vorgestellte Prozessbild gibt dem Leser einen definierten, agilen Prozess vor. Dies hat zum Einen den Vorteil, dass sich der Prozess „by the book“ einführen und etablieren lässt. Es birgt zum Anderen jedoch die Gefahr, dass sich dieser Prozess nach einer Einführung nicht mehr weiter entwickelt. Die Weiterentwicklung von Methoden und Prozessen auf die jeweiligen kulturellen und organisatorischen Bedürfnisse in einem Unternehmen oder Team ist aber einer der wichtigsten Grundpfeiler aller Agilität. Es ist also nicht nur wünschenswert, dass sich Prozesse regelmäßig ändern, sondern Pflicht, wenn man eine wirklich agile Organisation erzeugen möchte.

Die in den folgenden Kapiteln aufgeführten Referenzen zwischen Norm und Umsetzung können sehr leicht dafür genutzt werden, den Einfluß von anstehenden Prozessänderungen auf die Normkonformität bewerten zu können.

Nachweis der Normkonformität anderer Prozesse

Die beschriebene Systematik, welche zu der in diesem Buch dargestellten Prozesslandschaft geführt hat, kann sowohl dafür genutzt werden, Software-Life-Cycle-Prozesses vollständig neu zu definieren, als auch die Normkonformität bestehender Software-Life-Cycle-Prozesse nachzuweisen bzw. in Frage zu stellen.



Grundsätzlich muss deutlich gesagt werden, dass beide Ansätze einen nicht unerheblichen Aufwand mit sich bringen, der im regulierten Umfeld jedoch in jedem Fall durchgeführt werden muss.

Der Nachweis der Normkonformität durch den Abgleich mit einem bestehenden oder neuen Software-Life-Cycle-Prozess ist aufwändig, mit dieser Systematik jedoch effektiv machbar.

Überblick der Prozesse und Artefakte

Das nachfolgende Diagramm zeigt den gesamten Überblick aller Prozesse und Artefakte, welche aus normativen und agilen Aspekten benötigt werden.

Agile Prozesse:

- Acceptance Testing*
- Clean Code / SOLID Principles*
- Continuous Integration*
- Early Integration / Early Testing*
- Exploratory and Manual Testing*
- Integration Testing*
- Product Backlog Grooming*
- Release Planning Meeting*
- Scrum*
- Sprint Planning*
- Sprint Retrospective*
- Sprint Review*
- Test Automation*
- Test-Driven Development (TDD)*
- Unit Testing*
- Zero Bug Tolerance*

Nicht-agile Prozesse:

- Problem Communication*
- Quality Management Prozess*
- Risk Analysis and Management*
- Software Maintenance Process*
- Team Kick-Off Meeting*
- Usability Process*

Agile Artefakte:

Definition of Done
Definition of Ready
Iteration Notes (Sprint Notizen)
Product Backlog
Sprint Development Plan = Sprint Backlog
Team Charter / Working Agreements
User Story

Nicht-agile Artefakte:

Architecture Documentation
Linkable IDs
Release Notes
Software Development Plan
Software Maintenance Plan
Test Documentation
Versioning System

Im folgenden Hauptteil dieses Buches werden die einzelnen Praktiken und Artefakte im Detail beschrieben in jeweils eigenen Kapiteln. Die Struktur dieser einzelnen Kapitel ist folgendermaßen aufgebaut:

- **Beschreibung** - kurze Erklärung der Praktik oder des Artefaktes für das bessere Verständnis, welche Absicht dahintersteckt und welche Ergebnisse erwartet werden.
- **Verantwortliche/beteiligte Rollen** - Liste der üblicherweise beteiligten Rollen für eine bessere Einschätzung, welche Rollen in einem bestehenden Unternehmenskontext für den Einsatz der Praktik oder des Artefaktes in Frage kommen. Die in diesem Buch aufgeführten Rollen müssen nicht genau so übernommen werden, sie dienen eher dem Aufzeigen potentieller, sinnvoller Möglichkeiten.
- **Referenzen** - Liste von Büchern, Artikeln und Webseiten, welche sich mit der Praktik oder dem Artefakt detailliert auseinandersetzen.
- **Umgesetzte Anforderungen aus der Norm** - Liste der direkt aus dem Normtext abgeleiteten Anforderungen an den Software-Life-Cycle-Prozess⁵. Zu jeder Anforderung aus der Norm liegt eine Beschreibung bzw. Begründung vor, aus der hervorgeht, aus welchem Grund die Praktik oder das Artefakt geeignet ist, diese Anforderung umzusetzen und die Herstellung der Normkonformität herzustellen.

Folgende Spalten sind in den Tabellen enthalten:

- **§** - Paragraph der IEC 62304, in dem die Anforderung an den Software-Life-Cycle-Prozess zu finden ist. Dabei kann es sich sowohl um den tatsächlichen Paragraphen in der Norm handeln, als auch um die entsprechenden Anmerkungen in Annex B der Norm.

⁵ Die Originaltexte aus der Norm IEC 62304 sind in der gekürzten Ausgabe dieses Buches aus lizenzrechtlichen Gründen nicht enthalten.

- **SK** - Sicherheitsklassifizierung, für welche die Anforderung aus der Norm umzusetzen ist.
 - **Normtext** - Originaltext aus dem entsprechenden Paragraphen der IEC 62304 bzw. dessen Anmerkungen in Annex B oder eine davon abgeleitete Anforderung an den Software-Life-Cycle-Prozess.
 - **Herstellung der Normkonformität** - Beschreibung bzw. Begründung, aus welchem Grund die Praktik oder das Artefakt geeignet ist, diese Anforderung umzusetzen.
- **Verknüpfungen zu anderen Praktiken und Artefakten** - Querverweise zu anderen Praktiken und Artefakten in diesem Dokument, welche einen direkten Bezug zur beschriebenen Praktik oder dem Artefakt besitzen.

Prozesse, Meetings, Reviews

Acceptance Testing

Beschreibung

Akzeptanz-Tests sind ein grundlegender Bestandteil von (agilen) Anforderungen. Durch ihre Einhaltung wird sichergestellt, dass (a) die gewünschten Anwenderbedürfnisse befriedigt werden, (b) die umgesetzte Funktionalität durch geprüfte Schnittstellen integriert werden kann und (c) die Anforderung als "fertig" (done) betrachtet werden kann.

Die Basis der (idealerweise automatisierten) Akzeptanz-Tests sind definierte Akzeptanz-Kriterien, welche jeweils zu einzelnen Anforderungen und User Stories gehören. Akzeptanz-Kriterien sind nicht wie klassische Requirements hart beschrieben, sondern formulieren umgangssprachlich die Intention des Anwendernutzens⁶.

Akzeptanz-Kriterien definieren den Rahmen einer User Story und werden nach Fertigstellung einer Story dazu benötigt, den tatsächlichen Fertigstellungsgrad⁷ fest zu stellen hinsichtlich der ursprünglich formulierten Intention. Sie entstehen in Zusammenarbeit zwischen Product Owner und Entwicklungsteam, sowie idealerweise mit Beteiligung der Kunden bzw. Anwender des Produktes.

Als Hilfsmittel in der agilen Produktentwicklung sind Akzeptanz-Kriterien das geeignete Mittel, um in kurzen Feedbackzy-

⁶ Im medizintechnischen Umfeld kann es durchaus Fälle geben, in denen technische Schnittstellen zu Geräten und anderer Software z.B. mit harten Performance-Anforderungen spezifiziert werden müssen. Es spricht nichts dagegen, solche Abhängigkeiten mit Hilfe von Akzeptanz-Kriterien zu beschreiben.

⁷ Level of Done.

klen⁸ fest zu legen, in welche Richtung das gewünschte und benötigte Produkt entwickelt werden soll. Damit hat das Entwicklungs-Team eine Methodik an der Hand, um für sich selbst jederzeit beurteilen zu können, ob es das richtige Produkt entwickelt⁹.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team

Referenzen

- Lisa Crispin - Agile Testing
- http://en.wikipedia.org/wiki/Acceptance_testing
- <http://www.boost.co.nz/blog/agile/acceptance-criteria/>
- <http://guide.agilealliance.org/guide/acceptance.html>

⁸ Iterationen bzw. Sprints.

⁹ Die Beurteilung durch das Entwicklungs-Team kann sowohl manuell als auch maschinell durch Entwicklung entsprechender Test-Automation durchgeführt werden. Die Investition in eine gute Test-Automation zahlt sich bereits bei der Entwicklung kleiner Software-Systeme schnell aus, da der Aufwand manueller Akzeptanz-Tests kontinuierlich anwächst und damit nicht mehr vollständig während einer Iteration durchgeführt werden kann.

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.2	ABC	To demonstrate that the requirements have been implemented as desired, each requirement should be stated in such a way that objective criteria can be established to determine whether it has been implemented correctly.	Mit Hilfe von definierten Akzeptanz-Tests wird sichergestellt, dass das erstellte Software-Produkt die Bedürfnisse der Anwender befriedigt und die Anforderungen korrekt umgesetzt wurden.
§ 5.5	BC	ensure that SOFTWARE UNITS meet acceptance criteria	Durch Akzeptanztests wird sichergestellt, dass die entwickelte Software entsprechende Akzeptanzkriterien befriedigt.
§ 5.5	BC	establish acceptance criteria for SOFTWARE UNITS prior to integration into larger SOFTWARE ITEMS (risk control measures, interfaces, coding standards)	Mit Hilfe von Akzeptanz-Tests wird sichergestellt, dass eine in Software umgesetzte Anforderung abgekapselt funktioniert und damit für eine Integration in andere Strukturen bereit ist.
§ 5.6	ABC	In order to fully test a SOFTWARE PRODUCT both black and white box testing might be required.	Akzeptanztests entsprechen Black-Box-Tests.

Clean Code / SOLID Principles

Beschreibung

Clean Code ist ein Wertesystem zur Professionalisierung der Software-Entwicklung. In ihm sind viele Good Practices¹⁰ und grundlegende Prinzipien enthalten, welche durch die Erfahrungen in den letzten Jahrzehnten als gültig und korrekt betrachtet werden können.

Bekannter sind in erster Linie die fünf SOLID-Prinzipien:

- **SRP** - Single Responsibility Principle
Eine Klasse darf nur aus einem Grund geändert werden.
- **OCP** - Open Closed Principle
Eine Klasse muss offen sein für Erweiterungen, aber geschlossen gegen Modifikationen.
- **LSP** - Liskov Substitution Principle
Eine abgeleitete Klasse verhält sich immer so wie ihre Basis-klasse.

¹⁰ Ich höre immer wieder den Wunsch aus Teams und Organisationen, sie hätten gerne eine Liste mit „Best Practices“, die sie befolgen müssen, damit alles wie gewünscht gut wird. Diesem Wunsch möchte ich nicht nachkommen, da es den Lean-Prinzipien widerspricht, einen definierten Regelsatz vorzugeben, der die Lösung aller Probleme darstellt. Jedes Team und jede Organisation muss in ihrem ganz spezifischen Umfeld ihre eigenen „Best Practices“ entwickeln. Sie sind jedoch nicht ohne weiteres auf andere Teams und Organisationen transferierbar. In diesem Buch spreche ich deswegen nur von „Good Practices“, aus denen sich im Einzelfall „Best Practices“ empirisch ableiten können.

- **ISP** - Interface Segregation Principle
Clients werden nicht mit Details versehen, die sie nicht benötigen.
- **DIP** - Dependency Inversion Principle
Klassen hoher Ebenen sollen nicht von Klassen niedriger Ebenen abhängig sein, sondern beide sollen von Interfaces/ Abstraktionen abhängen. Interfaces sollen nicht von Details abhängen, sondern Details von Interfaces.

Der Nutzen von Clean Code zeigt sich im Software-Life-Cycle-Prozess ab dem Zeitpunkt, an dem die Maintenance-Phase des Produktes beginnt. Dies ist erfahrungsgemäß der längste Zeitraum im gesamten Life-Cycle. Wird das Clean Code-Ziel der intuitiven Verständlichkeit von Quellcode erreicht, kann in der Maintenance-Phase sehr viel Aufwand eingespart werden, in dem das Entwicklungs-Team ansonsten versuchen würde, sich den Quellcode, die darin verborgenen Strukturen und Zusammenhänge zu erarbeiten und nachzuvollziehen.

Verantwortliche/beteiligte Rollen

- Entwicklungs-Team

Referenzen

- http://de.wikipedia.org/wiki/Clean_Code
- <http://www.clean-code-developer.de/SOLID.ashx>
- <http://www.cleancoders.com/>
- Robert C. Martin - Clean Code

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 4	ABC	software engineering	Modernes Software-Engineering wird umgesetzt mit der Anwendung von Clean Code und SOLID-Prinzipien, TDD, Unit Tests, Design Patterns und vielen anderen Methoden, die hier nicht vollständig aufgeführt werden können.
§ 4	ABC	The software ARCHITECTURE should promote segregation of software items that are required for safe operation and should describe the methods used to ensure effective segregation of those SOFTWARE ITEMS.	Durch die korrekte Anwendung objekt-orientierter Prinzipien wie "Separation of Concerns" und "Single Responsibility Principle" entstehen lose gekoppelte Architekturen.
§ 5.4	ABC	define the level of detail of the SOFTWARE UNIT (can also be larger software items which are not decomposed)	Durch die Anwendung objekt-orientierter Prinzipien wird die Aufteilung von Software-Einheiten in weitere Einheiten getrieben.

Continuous Integration

Beschreibung

Kontinuierliche Integration beschreibt den Prozess des regelmäßigen, vollständigen Neubildens und Testens einer Anwendung.

Jeder Entwickler checkt seine Änderungen früh und regelmäßig (mindestens täglich) in eine Versionsverwaltung ein. Von dort aus wird das gesamte System neu gebaut und automatisch getestet, damit die Entwickler schnellstmöglich auf vorhandene Fehler im System aufmerksam gemacht werden können.

Die beiden großen Vorteile von Continuous Integration sind zum einen die Vermeidung hoher und teurer Integrationsaufwände am Ende eines Projektes und zum anderen die Möglichkeit, zu jedem beliebigen Zeitpunkt ein lauffähiges Produkt aus der Versionsverwaltung auschecken zu können.

Verantwortliche/beteiligte Rollen

- Entwicklungs-Team

Referenzen

- http://en.wikipedia.org/wiki/Continuous_integration
- <http://www.martinfowler.com/articles/continuousIntegration.html>
- <http://guide.agilealliance.org/guide/ci.html>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.1	BC	include or reference in the software development plan, a plan to integrate the SOFTWARE ITEMS (including SOUP) and perform testing during integration.	Konsequent angewandte Continuous Integration ist die Aktivität, durch welche eine frühzeitige Integration aller beteiligten Komponenten inklusive der dazugehörigen Integrationstests sichergestellt wird.
§ 5.6	BC	integrate the software units	Die einzelnen Software-Einheiten werden kontinuierlich integriert.

Verknüpfungen zu anderen Praktiken und Artefakten:

- **Versioning System** - Kontinuierliche Integration bedient sich eines Versionierungssystems, um jederzeit kleine Änderungen am System integrieren zu können.
- **Test Automation** - Kontinuierliche Integration benötigt eine funktionierende Test-Automatisierung, um ihren Nutzen entfalten zu können.

Early Integration / Early Testing

Beschreibung

Frühzeitige Integration und frühzeitiges Testen sind zum Einen Konzepte der kontinuierlichen Integration mit Test-Automati-on, zum Anderen können sie auch entwicklungsbegleitend manuell durchgeführt werden.

Im Extremfall können Anforderungen vor ihrer eigentlichen Umsetzung bereits getestet werden (Test-First Development).

Das darunterliegende Prinzip heißt "Schnelles Scheitern" (Fail Fast). Wir möchten es so schnell wie möglich wissen, falls unser Vorhaben nicht erfolgreich sein kann.

Durch die frühe und regelmäßige Integration mit externen Bestandteilen des Produktes aus anderen Teams erhalten wir eine schnelle Feedbackschleife, die uns dabei hilft, die Qualität, Architektur und Machbarkeit des Gesamtproduktes zu beurteilen. Wir können somit vermeiden, am Ende des Projektes hohe Integrationsaufwände investieren zu müssen.

Verantwortliche/beteiligte Rollen

- Entwicklungs-Team

Referenzen

- James Shore - The Art of Agile Development (Kapitel 13.2)
- <http://www.ibm.com/developerworks/rational/library/early-integration-testing-enables-agile-development/>
- <http://guide.agilealliance.org/guide/integration.html>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.7	BC	It is acceptable to test software requirements in earlier phases.	Anforderungen können bereits vor ihrer endgültigen Umsetzung getestet und integriert werden.

Verknüpfungen zu anderen Praktiken und Artefakten

- **Versioning System** - Frühe Integration bedient sich eines Versionierungssystems, um Änderungen am System sofort integrieren zu können.

Exploratory and Manual Testing

Beschreibung

Neben einer funktionierenden Test-Automation beschäftigen sich agile, entwicklungsbegleitende Tester mit explorativem und manuellem Testen des Systems.

Beim explorativem Testen wird das System in bestimmten Zielaspekten betrachtet. Es existiert kein vorher festgelegter Testplan, sondern dieser entsteht durch den Tester bei der eigentlichen Testarbeit, sich diesen Zielaspekten im System zu nähern und diese herauszufordern. Dabei entsteht ein Testplan, ein Testentwurf und die tatsächliche Testdurchführung.

Beim manuellen Testen werden vorher festgelegte Testpläne durchlaufen, die entweder aus klassischen Testentwürfen entstehen oder als Ergebnis des explorativen Testens als neue Teilstestpläne in den Gesamttestplan einfließen.

Der Vorteil des explorativen Testens gegenüber dem manuellen Testen ist die Flexibilität des Testers, sich unbekanntem, neuen oder veränderten Teilen des System intuitiv zu nähern und diese bezüglich ihrer Qualität zu „erforschen“.

Verantwortliche/beteiligte Rollen

- Entwicklungs-Team

Referenzen

- Lisa Crispin - Agile Testing
- <http://de.slideshare.net/codecentric/exploratory-testing-in-agileoverviewmeettheexpertselisabethhendrickson>
- <http://www.satisfice.com/articles/et-article.pdf>

- http://en.wikipedia.org/wiki/Exploratory_testing
- <http://guide.agilealliance.org/guide/exploratory.html>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.6	ABC	In order to fully test a SOFTWARE PRODUCT both black and white box testing might be required.	Exploratives Testen entspricht Black-Box-Tests.

Verknüpfung zu anderen Praktiken und Artefakten

- **Test Documentation** - Die benötigte Test Dokumentation wird mit den Ergebnissen der explorativen und manuellen Tests angereichert.

Integration Testing

Beschreibung

Integrations-Tests stellen sicher, dass sich die einzelnen Teile eines entwickelten System miteinander verbinden lassen und ein funktionierendes Ganzes entsteht.

Das Integrations-Testen befindet sich zwischen den Unit-Tests und der Validierung des Systems. Im agilen Context wollen wir das System innerhalb einer Iteration vollständig integriert haben mit allen Systemteilen, die in dieser Iteration entstanden sind. Dies führt zu der Notwendigkeit, Integrations-Tests zu jedem beliebigen Zeitpunkt durchführen zu können, was nichts anderes bedeutet, dass auch Integrations-Tests automatisiert werden müssen.

Verantwortliche/beteiligte Rollen

- Entwicklungs-Team

Referenzen

- <http://c2.com/cgi/wiki?IntegrationTest>
- http://en.wikipedia.org/wiki/Integration_testing
- <http://guide.agilealliance.org/guide/integration.html>
- Lisa Crispin - Agile Testing

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.6	BC	test the integrated SOFTWARE ITEMS in accordance with the integration plan and document the results	Die integrierten Software-Einheiten werden durch Integrationstests verifiziert. Durch Test-Automation kann eine Dokumentation der Testergebnisse erzeugt werden.
§ 5.6	BC	verify and record all aspects of the software integration in accordance with the integration plan	Die Software-Integration wird durch Integrationstests verifiziert.
§ 5.6	ABC	In order to fully test a SOFTWARE PRODUCT both black and white box testing might be required.	Integrations-Tests entsprechen Black-Box-Tests bezüglich der Schnittstellen einzelner Systemkomponenten und White-Box-Tests bezüglich der System-Architektur.
§ 5.6	ABC	Software integration and system testing can be performed in a simulated environment, on actual target hardware, or on the full MEDICAL DEVICE.	Integrations- sowie System-Tests können mit Simulationen, Prototypen und Mocks durchgeführt werden.

Problem Communication

Beschreibung

Ein Problem-Kommunikationsprozess wird dafür eingesetzt, Stakeholder, Anwender, Kunden und allgemein alle am Produkt interessierten Personen darüber zu informieren, welche Fehler und Probleme sich in dem Produkt befinden und beheben bzw. nicht länger benutzt werden sollten.

In diesem Buch wird nicht weiter auf die Möglichkeiten eingegangen, einen Problem-Kommunikationsprozess zu definieren und zu etablieren, da es sich hierbei um klassische Projektmanagement-Themen handelt, zu denen umfangreiche Referenzen und etablierte Methoden existieren wie z.B. Projektmarketing, Projektkommunikation oder Stakeholder-Management.

Verantwortliche/beteiligte Rollen

- Product Owner

Referenzen

- Schelle, Ottmann, Pfeiffer - ProjektManager (beinhaltet das gesamte Projektmanagement-Kompendium der GPM/IPMA)

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 6.2	ABC	inform users and regulators about any problem in released SOFTWARE PRODUCTS and the consequences of continued unchanged use	Probleme und deren Konsequenzen werden über einen "Problem Communication"-Prozess an Anwender kommuniziert.
§ 9	ABC	advise relevant parties of the existence of the problem	Einzubindende Stakeholder werden durch einen "Problem Communication"-Prozess über die Existenz eines Problems informiert.

Product Backlog Grooming

Beschreibung

Das Product Backlog Grooming Meeting dient dazu, dem Team ein sauberes, verständliches und bereites Product Backlog für mindestens die kommende Iteration verfügbar zu machen.

Die Inhalte des Product Backlogs werden auf Aktualität, Wichtigkeit, Verständnis, Schätzbarkeit und Nutzen geprüft und bei Bedarf entsprechend gesplittet, verfeinert, geändert oder entfernt. Es können auch neue Anforderungen bzw. User Stories entstehen. Grundsätzlich wird die Frage gestellt, ob das Product Backlog und seine enthaltenen Stories der Produktvision entsprechen und den richtigen Weg darstellen, diese Vision umzusetzen. Dabei legen wir den Detailfokus auf die nächsten 3 Sprints. Alle Backlog-Einträge, die sich weiter in der Zukunft befinden, sollen inhaltlich größer, allgemeiner bzw. abstrakter werden¹¹.

Im Product Backlog Grooming Meeting wird von Product Owner und Team bewertet, welche Änderungen sich am Produkt durch Anforderungen und Feedback aus dem Sprint Review (z.B. Änderungsanträge und User Stories) ergeben. Die Ent-

¹¹ Es ist wichtig, die Einträge des Backlogs nach unten hin immer größer, allgemeiner und abstrakter werden zu lassen. Zum Einen führt es dazu, dass das Team einen schnellen Überblick über das Backlog aufrecht erhalten kann und grob abschätzen kann, welche Themen noch umgesetzt werden sollen. Zum Anderen vermeiden wir dadurch, ein bei Projektbeginn bereits durchspezifiziertes Backlog zu erstellen, von dem wir dann nur noch schwer wegkommen. Die empirische Prozesskontrolle agiler Methoden macht sich genau das zum Vorteil, dass wir zu jedem zukünftigen Zeitpunkt flexibel agieren und reagieren können und nicht auf ein damals gültiges, heute jedoch veraltetes Backlog festgelegt sind.

scheidung darüber, welche Anforderungen umgesetzt werden sollen, liegt einzig und alleine beim Product Owner.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team
- Scrum Master

Referenzen

- <http://www.romanpichler.com/blog/product-backlog/grooming-the-product-backlog/>
- <http://www.scrumalliance.org/articles/339-how-to-hold-an-effective-backlog-grooming-session>
- <http://guide.agilealliance.org/guide/grooming.html>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.2	ABC	ensure that existing requirements, including SYSTEM requirements, are re-EVALUATED and updated as appropriate as a result of the software requirements analysis ACTIVITY	Im Product Backlog Grooming Meeting werden bestehende Anforderungen und User Stories im Product Backlog neu bewertet und aktualisiert.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.3	BC	specify the SYSTEM hardware and software necessary to support the proper operation of the SOUP item	Im Backlog Grooming werden die Anforderungen an Hardware und Software identifiziert, welche durch den Einsatz von SOUP notwendig werden.
§ 5.3	BC	specify functional and performance requirements for the SOUP item that are necessary for its intended use	Im Product Backlog Grooming werden die Funktionalitäten und Performance-Anforderungen an einzusetzende SOUP-Komponenten definiert.
§ 6.1	ABC	software maintenance plan includes use of the software problem resolution PROCESS for analysing and resolving problems arising after release of the MEDICAL DEVICE SOFTWARE	Im Product Backlog Grooming werden Probleme und Fehler von bereits releaster Software besprochen.

§	SK	Normtext	Herstellung der Normkonformität
§ 6.2	ABC	EVALUATE Each PROBLEM REPORT to determine how it affects the SAFETY of a released SOFTWARE PRODUCT and whether a change to the released SOFTWARE PRODUCT is needed to address the problem	Im Product Backlog Grooming Meeting werden Fehler- und Problem-Berichte bewertet und ihr Einfluß auf die Sicherheit bestehender Produkte und Systeme festgestellt, um notwendige Anforderungsänderungen im Product Backlog abzuleiten.
§ 6.2	ABC	EVALUATE and approve CHANGE REQUESTS which modify released SOFTWARE PRODUCTS	Der Product Owner läßt im Product Backlog Grooming Meeting neue und geänderte Anforderungen vom Team bewerten und erteilt deren Freigabe durch Aufnahme und Einsortierung in sein Product Backlog.
§ 6.2	BC	analyse each CHANGE REQUEST for its effect on the organization, released SOFTWARE PRODUCTS, and SYSTEMS with which it interfaces	Im Product Backlog Grooming Meeting werden neue und geänderte Anforderungen auf ihren Einfluß auf bestehende Produkte und Systeme untersucht.

§	SK	Normtext	Herstellung der Normkonformität
§ 6.3	ABC	use the software development PROCESS or an established maintenance PROCESS to implement the modifications	Im Product Backlog Grooming werden Änderungen an der Software analysiert, bewertet und ins Product Backlog einsortiert. Damit durchlaufen Änderungen den gleichen Entwicklungsprozess wie Anforderungen und User Stories.
§ 7.4	BC	analyse changes to the MEDICAL DEVICE SOFTWARE (including SOUP) to determine interference with existing risk control measures	Im Product Backlog Grooming Meeting durchlaufen neue und geänderte Anforderungen den Risikomanagement-Prozess, um sie mit bestehenden Risiko-Maßnahmen abzugleichen.
§ 7.4	ABC	analyse changes to the MEDICAL DEVICE SOFTWARE (including SOUP) to determine new potential hazards and risk control measures	Im Product Backlog Grooming Meeting durchlaufen neue und geänderte Anforderungen den Risikomanagement-Prozess, um neue Risiko-Maßnahmen abzuleiten.

§	SK	Normtext	Herstellung der Normkonformität
§ 9	ABC	approve and implement all CHANGE REQUESTS, observing the requirements of the change control PROCESS	Im Product Backlog Grooming werden notwendige Änderungen am System diskutiert und für die Entwicklung freigegeben.

Verknüpfungen zu anderen Praktiken und Artefakten

- **Product Backlog** - Im Product Backlog Grooming wird das Product Backlog kontinuierlich überarbeitet und angepasst.
- **User Story** - Im Product Backlog Grooming werden User Stories erstellt, verfeinert, abgeschätzt, verworfen und in einzelne User Stories aufgesplittet.

Quality Management Prozess

Beschreibung

Ein zertifiziertes Qualitätsmanagementsystem (z.B. nach ISO 13485) für medizinische Produkte wird dafür genutzt, die Normkonformität für Medizinproduktentwicklungen nachzuweisen und damit deren Marktfreigabe zu erwirken.

In diesem Buch wird auf das Thema Qualitätsmanagement nicht weiter eingegangen, da es sich auf einer anderen Ebene zum Software-Life-Cycle befindet und unabhängig von der jeweiligen Vorgehensweise umgesetzt werden sollte.

Verantwortliche/beteiligte Rollen

- Qualitätsmanager
- Scrum Master

Referenzen

- ISO 13485
- http://de.wikipedia.org/wiki/ISO_13485

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 4	ABC	use of a quality management system that complies with: ISO 13485; or a national quality management system standard; or a quality management system required by national regulation	Ein Qualitäts-Management-Prozess ist definiert und wird eingesetzt.

Release Planning Meeting

Beschreibung

Im Release Planning wird darüber entschieden, welcher Inhalt/Scope in die Produktentwicklung einfließen soll und welche Zeiträume und Meilensteine sich daraus ergeben. Ergebnis des Release Plannings ist ein initiales Product Backlog, mit dem das Team mit der Umsetzung beginnen kann.

Das Release Planning Meeting findet idealerweise vor oder nach einem Team-Kick-Off-Meeting statt.

Gibt es die Notwendigkeit, dieses Meeting regelmäßig durchzuführen¹², dann sollte es mit dem Product Backlog Grooming verschmolzen werden, um nicht ein weiteres, regelmäßiges Meeting einzuführen, welches die Teamkapazität reduzieren würde.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team

Referenzen

- <http://www.energizedwork.com/weblog/2006/04/release-planning.html>

¹² Die Ursachen für Notwendigkeiten dieser Art müssen vom ScrumMaster dringend untersucht und möglicherweise beseitigt werden. Es stecken oft organisatorische Unzulänglichkeiten hinter ständigen, negativen Veränderungen im Produktportfolio. Fehlende Vision, Aktionismus bei Problemen oder unorganisierte Wartungstaktik können grundlegende Ursachen für ständige, negative Veränderungen sein.

- <http://www.extremeprogramming.org/rules/planninggame.html>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.8	BC	ensure that the released SOFTWARE PRODUCT can be reliably delivered: replication, media labelling, packaging, protection, storage, and delivery.	Im Release Planning wird entschieden, welche Maßnahmen und Aktivitäten durchzuführen sind, um eine verlässliche Auslieferung der Software zu gewährleisten.
§ 6.1	ABC	software maintenance plan includes use of the software problem resolution PROCESS for analysing and resolving problems arising after release of the MEDICAL DEVICE SOFTWARE	Im Release Planning findet die Vorbewertung und Priorisierung statt für Probleme und Fehler von bereits releaster Software.
§ 6.3	ABC	release modified SOFTWARE SYSTEMS according to 5.8 (Software Release)	Das Release Planning unterscheidet nicht, ob ein Release aus neuen Anforderungen besteht oder aus Änderungen an einem bestehenden System.

§	SK	Normtext	Herstellung der Normkonformität
§ 6.3	ABC	Modifications may be released as part of a full re-release of a SOFTWARE SYSTEM or as a modification kit comprising changed SOFTWARE ITEMS and the necessary tools to install the changes as modifications to an existing SOFTWARE SYSTEM	Im Release Planning wird festgelegt, ob ein Release als vollständiges System-Release erstellt wird oder als Patch zu einem bestehenden System-Release.

Verknüpfung mit anderen Praktiken und Artefakten

- **Product Backlog** - Im Release Planning wird das Product Backlog initial erstellt bzw. in späteren Release Planning Meetings kontinuierlich überarbeitet und angepasst.

Risk Analysis and Management

Beschreibung

In einem Risiko-Management-Prozess werden Gefährdungen für Menschen und Umwelt durch Gebrauch eines Medizinproduktes analysiert, bewertet, durch Maßnahmen reduziert und entsprechend kontrolliert.

In der IEC 62304 nimmt das Thema Risiko-Management eine orthogonale Stellung zum gesamten Software-Life-Cycle-Prozess ein. Das bedeutet, dass sämtliche Aspekte des Life-Cycle-Prozesses vom Risiko-Management umgeben sind. Oder anders ausgedrückt: der Risiko-Management-Prozess findet während aller Entwicklungsaktivitäten und -phasen statt.

In diesem Buch wird nicht näher auf den eigentlichen Risiko-Management-Prozess und seine Umsetzung im Detail eingegangen. Die Anforderungen an den Risiko-Management-Prozess sind in der Norm ISO 14971 beschrieben und von dem jeweils eingesetzten Vorgehensmodell im Software-Life-Cycle-Prozess unabhängig.

Verantwortliche/beteiligte Rollen

- Qualitätsmanager
- Product Owner
- Entwicklungs-Team
- Scrum Master

Referenzen

- ISO 14971

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 1	ABC	perform MEDICAL DEVICE RISK MANAGEMENT that is compliant with ISO 14971	Risiko-Management wird in erster Linie im Sprint Planning durchgeführt.
§ 4	ABC	RISK MANAGEMENT should be applied as an integral part of the development PROCESS	Risiko-Management wird im Sprint Planning Meeting durchgeführt.
§ 4	ABC	assign to each SOFTWARE SYSTEM a software safety class (A, B, or C) according to the possible effects on the patient, operator, or other people resulting from a HAZARD to which the SOFTWARE SYSTEM can contribute	Im Sprint Planning wird jeder Anforderung und User Story (auf jeder Ebene) eine Sicherheitsklassifizierung zugeordnet durch die Anwendung der Risiko-Analyse.
§ 5.2	BC	include RISK CONTROL measures implemented in software for hardware failures and potential software defects in the requirements	Im Sprint Planning Meeting wird sichergestellt, dass alle Maßnahmen aus der Risiko-Analyse einer in diesem Sprint geplanten Anforderung in das Product Backlog als Anforderung einfließen.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.2	ABC	re-EVALUATE the MEDICAL DEVICE RISK ANALYSIS when software requirements are established and update it as appropriate	Im Sprint Planning Meeting durchlaufen alle Anforderungen eine Risiko-Analyse, um notwendige Maßnahmen abzuleiten.
§ 5.3	ABC	during all ACTIVITIES of the software ARCHITECTURE, the classification of the SOFTWARE SYSTEM and SOFTWARE ITEMS should be re-EVALUATED	Die Sicherheitsklassifizierung wird im Sprint Planning für alle umzusetzenden Software-Einheiten überdacht und gegebenenfalls neu vergeben.
§ 5.3	C	identify the segregation between SOFTWARE ITEMS that is essential to RISK CONTROL, and state how to ensure that the segregation is effective	Im Sprint Planning werden die Architektur und die Software-Einheiten der umzusetzenden Anforderungen definiert. Die identifizierten Software-Einheiten durchlaufen die Risiko-Analyse.
§ 5.7	BC	When changes are made during SOFTWARE SYSTEM testing, perform relevant RISK MANAGEMENT ACTIVITIES	Notwendige Änderungen an der Software, die durch System-Tests erkannt werden, durchlaufen während des Sprint Plannings die notwendige Risikoanalyse.

§	SK	Normtext	Herstellung der Normkonformität
§ 7.1	BC	document in the RISK MANAGEMENT FILE sequences of events that could result in a hazardous situation	Im Sprint Planning werden Workflows und sequentielle Abfolgen von Events in der Software durch die Risiko-Analyse auf Gefahrensituationen untersucht.
§ 7.1	BC	identify SOFTWARE ITEMS that could contribute to a hazardous situation identified in the MEDICAL DEVICE RISK ANALYSIS ACTIVITY of ISO 14971	Im Sprint Planning werden User Stories daraufhin untersucht, ob sie zu einer Gefahrensituation beitragen können, die in der Risiko-Analyse enthalten ist.
§ 7.1	BC	consider potential causes including hardware failures or other software defects that could result in unpredictable software operation	Im Sprint Planning werden für fehlerhaftes Verhalten der Software mögliche Ursachen in Hardware- und Software-Fehlern analysiert.
§ 7.1	BC	document in the RISK MANAGEMENT FILE potential causes of the SOFTWARE ITEM contributing to a hazardous situation	Im Sprint Planning werden die umzusetzenden Software-Einheiten durch die Risiko-Analyse auf Gefahrensituationen untersucht.

§	SK	Normtext	Herstellung der Normkonformität
§ 7.1	BC	consider potential causes including software defects in the identified SOFTWARE ITEM functionality	Im Sprint Planning werden die umzusetzenden Anforderungen und deren abgeleitete Software-Einheiten durch die Risiko-Analyse auf mögliche Gefährdungen hin untersucht.
§ 7.1	BC	consider potential causes including failure or unexpected results from SOUP	Im Sprint Planning werden mögliche Ursachen für Fehlverhalten von SOUP durch die Risiko-Analyse identifiziert.
§ 7.1	ABC	all device hazardous situations can be identified in parallel to definition of the software ARCHITECTURE	Im Sprint Planning wird die Risiko-Analyse der umzusetzenden Architektur und ihrer Software-Einheiten durchgeführt.
§ 7.1	BC	evaluate SOUPs and their list of known anomalies that could lead to a hazardous situation	Im Sprint Planning werden SOUPs und ihre bekannten Fehler/Abweichungen durch die Risiko-Analyse betrachtet.

§	SK	Normtext	Herstellung der Normkonformität
§ 7.1	BC	consider potential causes including reasonably foreseeable misuse	In der Risiko-Analyse während des Sprint Planning Meetings werden alle Anforderungen auf möglichen Mißbrauch untersucht und entsprechende Maßnahmen abgeleitet.
§ 7.2	BC	define and document risk control measures for each potentially hazardous software item	Im Sprint Planning werden zu jeder Software-Einheit mit möglichen Gefahrensituationen entsprechende Maßnahmen identifiziert und festgehalten.
§ 7.2	BC	assign a software safety class to the SOFTWARE ITEM based on the possible effects of the HAZARD that the RISK CONTROL measure is controlling	Im Sprint Planning wird der zu erstellenden Software-Einheit eine Sicherheitsklassifizierung zugewiesen (A, B, C), welche weitere Details im Entwicklungsprozess bestimmt.
§ 7.2	BC	include the RISK CONTROL measure in the software requirements	Im Sprint Planning Meeting wird sichergestellt, dass alle Maßnahmen aus der Risiko-Analyse einer in diesem Sprint geplanten Anforderung in das Product Backlog als Anforderung einfließen.

§	SK	Normtext	Herstellung der Normkonformität
§ 9	ABC	EVALUATE the problem's relevance to SAFETY using the software RISK MANAGEMENT PROCESS	Im Sprint Planning Meeting werden Probleme und Fehler auf sicherheitsrelevante Aspekte mit Hilfe der Risikoanalyse untersucht.

Scrum

Beschreibung

Scrum ist ein agiles Management-Framework für die empirische Prozesskontrolle. Es kann ideal eingesetzt werden bei jeder Art von Produktentwicklung.

Scrum basiert auf den drei Grundpfeilern

- **Transparenz** - für jeden Beteiligten an der Produktentwicklung müssen die verwendeten Prozesse, Inhalte, Anforderungen, Ziele, Visionen, etc. erkennbar sein, damit ein gemeinsames Verständnis entstehen kann und zielgerichtete Teamarbeit möglich wird.
- **Inspektion/Überprüfung** - alle Ergebnisse und Artefakte, sowie die Art und Weise der Zusammenarbeit werden in regelmäßigen Abständen betrachtet und hinsichtlich der beabsichtigten Zielerreichung überprüft.
- **Adaption/Anpassung** - die Erkenntnisse aus der Inspektion/Überprüfung fließen ein in konkrete Veränderungen am Prozess oder der gewünschten Ziele und Inhalte.

Jedes Meeting in Scrum beinhaltet die Aspekte von Inspektion und Adaption, somit ist die empirische Prozesskontrolle bzw. ein kontinuierlicher Verbesserungsprozess Grundbestandteil von Scrum.

In diesem Buch werden die einzelnen Bestandteile von Scrum beschrieben. Es kann jedoch die Lektüre grundlegender Scrum-Bücher nicht ersetzen. Für einen Einstieg in die Thematik sind die unten aufgelisteten Referenzen geeignet.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team
- Scrum Master

Referenzen

- <http://www.scrum.org/Scrum-Guides>
- <http://www.scrum.org/Resources/What-is-Scrum/>
- Ken Schwaber - Agiles Projektmanagement mit Scrum
- Roman Pichler - Scrum

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 1	ABC	Any waterfall, incremental, and evolutionary life cycle may be used.	Scrum ist ein gültiges Prozess-Framework für den Software-Life-Cycle.

Software Maintenance Process

Beschreibung

Der Software Maintenance-Prozess (Software Wartungsprozess) beschreibt alle Aktivitäten, die für die Änderung von sich im Markt befindlichen Medizinprodukten notwendig sind. Dazu gehört das Einholen und Bewerten von Anwender-Feedback sowie ein installierter Fehlerbehebungsprozess.

Die in diesem Buch beschriebenen Praktiken und Artefakte bilden fast den vollständigen Software Maintenance-Prozess ab, bis auf das explizite Einholen von Feedback bezüglich bereits releaster Produkte. Während der Entwicklung wird diese Anforderung mit Hilfe des Sprint Review Meetings abgedeckt, nach Projektabschluss werden andere Methoden benötigt.

In diesem Buch wird nicht näher auf die Art und Weise eingegangen, wie dieses Feedback von Anwendern und internen Mitarbeitern eingeholt werden kann, da dies unabhängig von dem jeweils eingesetzten Vorgehensmodell im Software-Life-Cycle-Prozess geschehen muss.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team

Referenzen

- https://www.ibm.com/developerworks/mydeveloperworks/blogs/c914709e-8097-4537-92ef-8982fc416138/entry/march_10_2012_11_27_am8?lang=en

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 6.2	ABC	monitor feedback on released SOFTWARE PRODUCT from users	Über den Software Maintenance Prozess erfolgt das Einholen des Feedbacks zu re-leaster Software von tatsächlichen Anwendern.
§ 6.2	ABC	monitor feedback on released SOFTWARE PRODUCT from inside its own organization	Über den Software Maintenance Prozess erfolgt das Einholen des Feedbacks zu re-leaster Software innerhalb der eigenen Organisation.

Sprint Planning

Beschreibung

Im Sprint Planning Meeting einigen sich der Product Owner und das Team auf ein Sprint-Ziel und das dafür umzusetzende Sprint Backlog. Dieses wird aus dem Product Backlog gepullt, abhängig von den dort gegebenen Prioritäten.

Ein Eintrag im oberen Teil des Product Backlog hat idealerweise folgende Eigenschaften:

- klein genug, damit das Team ihn innerhalb eines Sprints umsetzen kann
- geschätzt, damit die Diskussion über den Inhalt und die Intention des Eintrages stattgefunden hat
- unabhängig¹³, damit das Team sofort mit der Umsetzung beginnen kann
- klar definierte Akzeptanzkriterien, damit das Team weiß, wann die Umsetzung als fertig betrachtet werden kann

Diese Eigenschaften finden wir im Artefakt „Definition of Ready“ wieder, welches genau dafür sorgt, dass diese Eigenschaften vor dem Sprint Planning Meeting erarbeitet werden.

Für die einzelnen Einträge des Sprint Backlogs werden im zweiten Teil des Sprint Plannings vom Team konkrete Architektur- und Umsetzungsaufgaben identifiziert und geplant.

¹³ Unabhängigkeit ist nicht unter allen Umständen herstellbar. Wenn sich Abhängigkeiten zwischen Einträgen im Product Backlog ergeben, dann müssen diese Einträge so angeordnet werden, dass sie in einer sinnvoll umsetzbaren Reihenfolge vom Team gepullt werden können. Jeff Sutherland spricht mittlerweile nicht mehr von der Eigenschaft „unabhängig (independant)“, sondern von „sofort umsetzbar (immediately actionable)“.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team
- Scrum Master

Referenzen

- <http://www.mountaingoatsoftware.com/scrum/sprint-planning-meeting>
- <http://www.leadingagile.com/2012/08/simple-cheat-sheet-to-sprint-planning-meeting/>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.2	ABC	software requirements: are expressed in terms that avoid ambiguity	Im Sprint Planning Meeting wird sichergestellt, dass über die Anforderungen ein einheitliches Verständnis bei allen Beteiligten herrscht.
§ 5.2	ABC	ensure that existing requirements, including SYSTEM requirements, are re-EVALUATED and updated as appropriate as a result of the software requirements analysis ACTIVITY	Im Sprint Planning Meeting werden bestehende Anforderungen und User Stories im Product Backlog neu bewertet und aktualisiert.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.2	ABC	software requirements: do not contradict one another	Im Sprint Planning Meeting wird vom Team festgestellt, ob sich Anforderungen im Product Backlog widersprechen.
§ 5.3	BC	specify functional and performance requirements for the SOUP item that are necessary for its intended use	Im Sprint Planning werden für benötigte SOUP-Komponenten die Funktionalitäten und Performance-Anforderungen festgelegt.
§ 5.3	ABC	define the major structural components of the software, their externally visible properties, and the relationship among them	Die grundlegenden Software-Strukturen und Architektur-Entscheidungen können im Sprint Planning getroffen werden.
§ 5.3	BC	develop and document an ARCHITECTURE for the interfaces between the SOFTWARE ITEMS and the components external to the SOFTWARE ITEMS (both software and hardware), and between the SOFTWARE ITEMS	Im Sprint Planning werden die Architektur und die Schnittstellen der umzusetzenden Software-Einheiten sowie der einzubindenden externen Software-Einheiten festgelegt.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.4	BC	refine the SOFTWARE ITEMS and interfaces defined in the ARCHITECTURE to create SOFTWARE UNITS and their interfaces	Im Sprint Planning werden die umzusetzenden Anforderungen in eine Architektur überführt und diese wiederum in Software-Einheiten runtergebrochen.
§ 5.5	BC	establish acceptance criteria for SOFTWARE UNITS prior to integration into larger SOFTWARE ITEMS (risk control measures, interfaces, coding standards)	Im Sprint Planning Meeting wird sichergestellt, dass zu jeder Anforderung Akzeptanzkriterien für die Integration in andere Strukturen erstellt werden.
§ 5.7	BC	It is acceptable to combine integration testing and SOFTWARE SYSTEM testing into a single plan.	Im Sprint Planning erfolgt die Planung der Integrations-Tests sowie der System-Tests.
§ 6.1	ABC	software maintenance plan includes use of the software problem resolution PROCESS for analysing and resolving problems arising after release of the MEDICAL DEVICE SOFTWARE	Im Sprint Planning finden die notwendigen Risiko-Analysen sowie die Umsetzungsplanung statt für Probleme und Fehler von bereits releaster Software.

§	SK	Normtext	Herstellung der Normkonformität
§ 7.1	BC	consider potential causes including incorrect or incomplete specification of functionality	Spätestens im Sprint Planning Meeting werden Anforderungen auf ihre korrekte und vollständige Definition geprüft.
§ 7.2	BC	develop the SOFTWARE ITEM implementing a RISK CONTROL measure in accordance with Clause 5 (Software Development Process)	Im Sprint Planning werden Software-Einheiten, die eine Risiko-Maßnahme umsetzen, genauso behandelt wie alle anderen Anforderungen und befolgen den definierten Entwicklungsprozess.
§ 8.2	ABC	identify and perform any ACTIVITY that needs to be repeated as a result of the change, including changes to the software safety classification of SOFTWARE SYSTEMS and SOFTWARE ITEMS	Im Sprint Planning Meeting werden vom Team alle Tasks identifiziert, welche für die Umsetzung einer Änderung (erneut) durchgeführt werden müssen.
§ 9	ABC	investigate the problem and if possible identify the causes	Im Sprint Planning werden die Fehler- und Problembereiche analysiert.

Verknüpfung mit anderen Praktiken und Artefakten

- **Sprint Development Plan = Sprint Backlog** - Im Sprint Planning Meeting wird das Sprint Backlog bzw. der Sprint Development Plan festgelegt.
- **Iteration Notes (Sprint Notizen)** - Im Sprint Planning werden die Iteration Notes (Sprint Notizen) angefangen.
- **Architecture Documentation** - Im Sprint Planning Meeting wird in der zweiten Phase (Sprint Planning 2) die Architektur der umzusetzenden Anforderungen besprochen, skizziert und festgehalten.

Sprint Retrospective

Beschreibung

In der Sprint Retrospektive hat das Team die Möglichkeit, Verbesserungsmaßnahmen für die eigenen Prozesse und Strukturen zu identifizieren und zu beschließen. Es geht dabei nicht darum, einen Schuldigen zu identifizieren für die Dinge, die nicht funktioniert haben. Vielmehr ist die Zielsetzung einer Retrospektive, aus der letzten Iteration konkrete Maßnahmen und Veränderungen abzuleiten, die das Team sofort umsetzen kann.

Die Moderation einer Retrospektive erfordert sehr viel Soft Skills vom Scrum Master. Empathie, Mediation, konstruktiver Umgang mit Konflikten, sowie die Fähigkeit, Entscheidungsprozesse im Team zu unterstützen, sind nur einige Aspekte, die ein guter Moderator mitbringen muss.

Auf die vielen Möglichkeiten und Aktivitäten, mit denen eine Retrospektive gestaltet werden kann, wird in diesem Buch nicht weiter eingegangen. Der interessierte Leser findet in den unten aufgeführten Referenzen entsprechende Literatur zu dem Thema.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team
- Scrum Master

Referenzen

- Norman Kerth - Project Retrospectives
- Esther Derby, Diana Larsen - Agile Retrospectives

- Patrick Kua - The Retrospectives Handbook (<https://leanpub.com/the-retrospective-handbook>)
- <http://guide.agilealliance.org/guide/heartbeatretro.html>

Umgesetzte Anforderungen aus der Norm

Keine direkt umzusetzenden Anforderungen aus der Norm.

Verknüpfung mit anderen Praktiken und Artefakten

- **Team Charter / Working Agreements:** Der Team Charter und die Working Agreements können vom Team im Sprint Retrospective Meeting an veränderte Situationen und gegebene Notwendigkeiten angepasst werden.
- **Definition-of-Ready:** Die Definition-of-Ready kann vom Team im Sprint Retrospective Meeting an veränderte Situationen und gegebene Notwendigkeiten angepasst werden.
- **Definition-of-Done:** Die Definition-of-Done kann vom Team im Sprint Retrospective Meeting an veränderte Situationen und gegebene Notwendigkeiten angepasst werden.
- **Software Development Plan:** Der Software Development Plan kann vom Team im Sprint Retrospective Meeting an veränderte Situationen und gegebene Notwendigkeiten angepasst werden.

Sprint Review

Beschreibung

Im Sprint Review Meeting wird vom Team vorgestellt, was im letzten Sprint alles geschehen ist und welche User Stories bzw. Anforderungen umgesetzt sind. Ziel des Sprint Review Meetings ist es, Feedback von allen beteiligten Stakeholdern zu erhalten und auf dieser Basis das Product Backlog für die kommenden Sprints anzupassen.

Es ist ein weitverbreiteter Irrtum, dass das Sprint Review Meeting dafür genutzt werden soll, die umgesetzten Stories und Anforderungen final zu beurteilen, abzunehmen und frei zu geben. Dies soll nach Möglichkeit bereits während des Sprints geschehen, sobald die einzelnen Stories und Anforderungen vom Team umgesetzt wurden. Auch hier gilt das Prinzip des frühen Feedbacks. Das Sprint Review Meeting ist gewissermaßen die letzte Möglichkeit für den Product Owner zur Abnahme und Freigabe, birgt jedoch immer die Gefahr, dass Fehler und Mißverständnisse zu spät entdeckt werden und es damit nicht mehr möglich ist, diese noch im Sprint zu beseitigen.

Ein weiterer Irrtum besteht darin, das Sprint Review Meeting als reine Demonstration der umgesetzten Features zu nutzen. Eine Demo des entstandenen Produktinkrements ist zweifellos ein wichtiger Bestandteil des Sprint Review Meetings, es geht jedoch viel mehr darum, das Feedback von echten Anwendern und Stakeholdern einzuholen.

Folgende Fragestellungen sollen im Sprint Review Meeting erörtert werden:

- Sind wir auf dem richtigen Weg mit unserer Produktentwicklung?
- Hilft das, was wir erzeugt haben, wirklich dem Anwender?

- Welche (neuen) Erkenntnisse haben die Anwender und Stakeholder bezüglich der weiteren Produktgestaltung gewonnen?
- Welche Teile des kommenden Product Backlogs sind noch wichtig, welche sind obsolet?

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team
- Scrum Master
- Stakeholder

Referenzen

- <http://www.mountaingoatsoftware.com/scrum/sprint-review-meeting>
- <http://www.scrumcrazy.com/Tips+for+a+Good+Sprint+Review>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 1	ABC	All PROCESS outputs should be maintained in a consistent state	Im Sprint Review wird die Konsistenz geprüft zwischen der erzeugten Software, ihrer Anforderungen und weiteren Ergebnissen bzw. Inputs.

§	SK	Normtext	Herstellung der Normkonformität
§ 1	ABC	before any MEDICAL DEVICE SOFTWARE is released, all PROCESS outputs should be consistent with each other and all dependencies between PROCESS outputs explicitly or implicitly required by this standard should be observed	Im Sprint Review werden alle Ergebnisse des gesamten Software-Life-Cycles auf gegenseitige Konsistenz geprüft.
§ 4	ABC	demonstrate the ability to provide MEDICAL DEVICE SOFTWARE that consistently meets customer requirements	Im Sprint Review Meeting wird festgestellt, ob das erzeugte Software-Produkt die Bedürfnisse und Anforderungen der Anwender befriedigt.
§ 4	ABC	demonstrate the ability to provide MEDICAL DEVICE SOFTWARE that consistently meets applicable regulatory requirements	Im Sprint Review Meeting wird festgestellt, ob die Entwicklung des erzeugten Software-Produkts den normativen Anforderungen entspricht.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.2	ABC	ensure that existing requirements, including SYSTEM requirements, are re-EVALUATED and updated as appropriate as a result of the software requirements analysis ACTIVITY	Die Ergebnisse und Erkenntnisse aus dem Sprint Review Meeting fließen als Feedback direkt in die Anforderungen ein und können im anschließenden Sprint Planning Meeting bedacht werden.
§ 5.4	C	verify and document that the software detailed design implements and follows the software ARCHITECTURE	Im Sprint Review wird präsentiert, dass das umgesetzte Software-Design der Software-Architektur entspricht.
§ 5.5	BC	ensure that SOFTWARE UNITS meet acceptance criteria	Im Sprint Review zeigt das Team, welche Akzeptanzkriterien in Form von Testfällen die Software erfolgreich durchlaufen hat.
§ 5.6	BC	EVALUATE the integration test procedures for correctness	Im Sprint Review werden die Integrations-tests auf Korrektheit geprüft.
§ 5.7	BC	verify that the VERIFICATION strategies and the test procedures used are appropriate	Im Sprint Review wird über Verifikation und Testen berichtet.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.7	BC	verify that all software requirements have been tested or otherwise VERIFIED	Im Sprint Review Meeting berichtet das Team über die durchgeführten Tests aller umgesetzten Anforderungen.
§ 5.7	BC	verify that test results meet the required pass/fail criteria	<p>Im Sprint Review werden die umgesetzten Anforderung auf Basis ihrer Akzeptanzkriterien abgenommen und freigegeben.</p> <p>Hinweis: das Sprint Review Meeting ist nicht dafür gedacht, die umgesetzten Features durch den Product Owner abnehmen zu lassen. Es stellt nur den letzten Zeitpunkt dar, an dem dies geschehen kann. Sinnvollerweise werden Features bereits während des Sprints direkt nach der Umsetzung vom Product Owner abgenommen.</p>

§	SK	Normtext	Herstellung der Normkonformität
§ 5.7	BC	verify that SOFTWARE SYSTEM test procedures trace to software requirements	Im Sprint Review Meeting präsentiert das Team die erzeugten und durchgeführten Tests zu den umgesetzten Anforderungen. (Es werden keine Tests erstellt, zu denen keine Anforderungen existieren.)
§ 5.7	ABC	test coverage of requirements, RISK CONTROL, usability, and test types (e.g., fault, installation, stress) should be demonstrated and documented	Im Sprint Review werden die entwickelten und durchlaufenen Tests, sowie die Test-Abdeckung demonstriert.
§ 5.8	BC	ensure that software VERIFICATION has been completed and the results EVALUATED before the software is released	Im Sprint Review werden die umgesetzten Anforderungen evaluiert und freigegeben, bevor sie in ein Software-Release gelangen können.
§ 7.3	BC	evaluate implemented risk control measures to identify new possible hazards	Im Sprint Review werden umgesetzte Risiko-Maßnahmen auf neue mögliche Gefahrensituationen hin untersucht.

§	SK	Normtext	Herstellung der Normkonformität
§ 7.3	BC	verify the implementation of each RISK CONTROL measure	Im Sprint Review wird die Umsetzung aller Risiko-Maßnahmen berichtet und verifiziert.
§ 9	ABC	perform analysis to detect trends in PROBLEM REPORTS	Im Sprint Review werden die Problem- und Fehlermeldungen auf Trends und Tendenzen untersucht.
§ 9	ABC	verify resolutions to determine whether additional problems have been introduced	Im Sprint Review wird berichtet, welche neuen Probleme und Fehler gefunden wurden.
§ 9	ABC	verify resolutions to determine whether CHANGE REQUESTS have been implemented in the appropriate SOFTWARE PRODUCTS and ACTIVITIES	Im Sprint Review wird berichtet, welche Änderungen (Change Requests) an der Software durchgeführt wurden.
§ 9	ABC	verify resolutions to determine whether problem has been resolved and the PROBLEM REPORT has been closed	Im Sprint Review wird berichtet, welche Fehler und Probleme behoben wurden.

§	SK	Normtext	Herstellung der Normkonformität
§ 9	ABC	verify resolutions to determine whether adverse trends have been reversed	Im Sprint Review wird berichtet, ob Problemlösungen dazu geführt haben, nachteilige Entwicklungen umzukehren.

Verknüpfung mit anderen Praktiken und Artefakten

- **Iteration Notes (Sprint Notizen)** - Im Sprint Review werden die Iteration Notes (Sprint Notizen) geschrieben und fertiggestellt.
- **Release Notes** - Im Sprint Review werden die Release Notes inkrementell erweitert.

Team Kick-Off Meeting

Beschreibung

Im Team Kick-Off-Meeting einigt sich das Team auf die grundlegenden Projektziele, die anzuwendenden Vorgehensweisen und Methoden, sowie die Art und Weise der eigenen Zusammenarbeit. In diesem Meeting werden der initiale Software-Entwicklungsplan und der Software-Maintenance-Plan entworfen.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team
- Scrum Master

Referenzen

- http://www.mindtools.com/pages/article/newTMM_95.htm

Umgesetzte Anforderungen aus der Norm

Keine direkt umzusetzenden Anforderungen aus der Norm.

Verknüpfung mit anderen Praktiken und Artefakten

- **Definition-of-Done** - Die Definition-of-Done wird initial in einem Team Kick-Off Meeting festgelegt.
- **Definition-of-Ready** - Die Definition-of-Ready wird initial in einem Team Kick-Off Meeting festgelegt.

- **Software Entwicklungsplan** - Der Software Entwicklungsplan wird initial in einem Team Kick-Off Meeting festgelegt.
- **Software-Maintenance-Plan** - Der Software Maintenance Plan wird initial in einem Team Kick-Off Meeting erstellt.
- **Team Charter/Working Agreements** - Ein Team Charter und die Working Agreements eines Teams werden initial in einem Team Kick-Off Meeting festgelegt.

Test Automation

Beschreibung

Eine Test-Automation ist die Basis für kontinuierliche Integration und ermöglicht eine effiziente Verifikation nach jeder kleinen Änderung am System.

Oft wird die Frage gestellt, ob wirklich jeder Test automatisiert werden soll. Die gültige Antwort darauf lautet: jeder Test, der mehr als ein mal durchgeführt werden wird, sollte automatisiert werden.

Durch Test-Automation wird ein Sicherheitsnetz hergestellt, welches uns ermöglicht, jederzeit ein lauffähiges Gesamtsystem herstellen zu können. Die gewonnene Sicherheit meldet uns Fehler, die sich durch Seiteneffekte in Schnittstellen und Funktionalitäten eingeschlichen haben. Diese Fehler können dann sofort beseitigt werden, so dass wir mit einem funktionierenden Gesamtsystem weiter arbeiten können.

Verantwortliche/beteiligte Rollen

- Entwicklungs-Team

Referenzen

- Lisa Crispin - Agile Testing
- <http://xprogramming.com/articles/automatedtesting/>
- <http://xprogramming.com/blog/automating-story-tests/>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.5	BC	ensure that SOFTWARE UNITS meet acceptance criteria	Durch Test-Automation wird sichergestellt, dass die Software immer wieder die definierten Akzeptanzkriterien in Form von automatisierten Tests erfolgreich durchläuft.
§ 5.5	BC	perform the SOFTWARE UNIT VERIFICATION and document the results	Durch Test-Automation können die Ergebnisse aller Unit-Tests als Dokumentation erzeugt werden.
§ 5.6	BC	conduct REGRESSION TESTING appropriate to demonstrate that defects have not been introduced into previously integrated software	Regressionstests vorhandener Software-Versionen können durch Test-Automation jederzeit sicherstellen, dass keine Fehler im Software-System entstanden sind.
§ 5.6	BC	retain sufficient records to permit the test to be repeated	Durch Test-Automation können bestehende Testfälle jederzeit wiederholt werden.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.6	BC	document the test result (pass/fail and a list of ANOMALIES)	Mit einer Test-Automatation kann durch die Durchführung aller vorhandenen Tests eine Dokumentation der Test-Resultate erzeugt werden.
§ 5.7	BC	verify that all software requirements have been tested or otherwise VERIFIED	Mit Hilfe einer umfassenden Test-Automatation kann sichergestellt und nachgewiesen werden, dass alle Anforderungen getestet und verifiziert sind.
§ 5.7	BC	establish and perform a set of tests, expressed as input stimuli, expected outcomes, pass/fail criteria and procedures, for conducting SOFTWARE SYSTEM testing, such that all software requirements are covered	Durch Test-Automatation können die Testfälle aller Software-Anforderungen jederzeit durchgeführt werden.
§ 5.7	ABC	When a change is made to a SOFTWARE SYSTEM, REGRESSION TESTING should be determined, planned and documented.	Durch Test-Automatation kann das gesamte Software-System nach einer Änderung vollständig mit allen vorhandenen Testfällen geprüft werden.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.7	BC	When changes are made during SOFTWARE SYSTEM testing, conduct testing appropriate to demonstrate that unintended side effects have not been introduced	Durch Test-Automation kann das gesamte Software-System nach einer Änderung vollständig mit allen vorhandenen Testfällen geprüft werden.
§ 5.7	BC	When changes are made during SOFTWARE SYSTEM testing, repeat tests, perform modified tests or perform additional tests, as appropriate, to verify the effectiveness of the change in correcting the problem	Durch Test-Automation können alle vorhandenen Testfälle des Software-Systems nach Änderungen jederzeit durchgeführt werden.
§ 8.2	ABC	verify the change, including repeating any VERIFICATION that has been invalidated by the change	Durch Test-Automation kann das Software-System jederzeit nach einer Änderung verifiziert werden.

Verknüpfung mit anderen Praktiken und Artefakten

- **Test Documentation** - Die benötigte Test Dokumentation wird mit Hilfe der Test Automation bei Bedarf auf Knopfdruck automatisch erzeugt.

- **Continuous Integration** - Kontinuierliche Integration benötigt eine funktionierende Test-Automatisierung, um ihren Nutzen entfalten zu können.

Test-Driven Development (TDD)

Beschreibung

Test-Driven Development (TDD) ist eine Entwicklungspraktik aus dem Extreme Programming (XP). Durch den angewendeten Test-First-Ansatz wird sichergestellt, dass (a) sämtlicher Produktivcode mit Tests versehen ist und (b) nur soviel Produktivcode entsteht, wie auch tatsächlich benötigt wird.

Die Vorteile von Test-Driven Development (TDD) sind mindestens folgende:

- Deutlich geringere Fehlerraten während und nach der Entwicklung eines Systems.
- Durchgängig testbarer Code entsteht durch die Anwendung von Test-Driven Development (TDD), so dass Testbarkeit nicht nachträglich und umständlich eingebaut werden muss.
- Ein jederzeit lauffähiges System entsteht automatisch.
- Eine ideale Entwicklerdokumentation entsteht durch die geschriebenen Tests, da diese Beispielimplementierungen für die Schnittstellen im Code bereit stellen.

Die anfängliche Investition in Test-Driven Development (TDD) zahlt sich sehr schnell durch die gewonnene Sicherheit und die geringeren Fehlerraten aus.

Verantwortliche/beteiligte Rollen

- Entwicklungs-Team

Referenzen

- Kent Beck - Test-Driven Development

- <http://butunclebob.com/ArticleS.UncleBob.TheThreeRulesOfTdd>
- <http://c2.com/cgi/wiki?TestDrivenDevelopment>
- <http://guide.agilealliance.org/guide/tdd.html>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.3	BC	transform the requirements for the MEDICAL DEVICE SOFTWARE into a documented ARCHITECTURE that describes the software's structure	Durch die Anwendung von Test-Driven Development (TDD) entsteht während der Umsetzung von Anforderungen eine funktionierende Software-Architektur.
§ 5.4	C	develop and document a detailed design for each SOFTWARE UNIT of the SOFTWARE ITEM	Durch die Anwendung von Test-Driven Development (TDD) entsteht ein "Design by Contract" über die Schnittstellenspezifikation aller Software-Einheiten in Form von ausführbaren Tests gegen diese Schnittstellen.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.4	C	develop and document a detailed design for any interfaces between the SOFTWARE UNIT and external components (hardware or software), as well as any interfaces between SOFTWARE UNITS	Durch die Anwendung von Test-Driven Development (TDD) entsteht die Schnittstellenspezifikation aller Software-Einheiten in Form von ausführbaren Tests gegen diese Schnittstellen.
§ 5.4	ABC	define the level of detail of the SOFTWARE UNIT (can also be larger software items which are not decomposed)	Durch Test-Driven Development (TDD) wird direkt bei der Umsetzung von Software-Einheiten entschieden, wann und wie diese weiter untergliedert werden in einzelne Einheiten.
§ 5.4	BC	refine the SOFTWARE ITEMS and interfaces defined in the ARCHITECTURE to create SOFTWARE UNITS and their interfaces	Durch den Einsatz von Test-Driven Development (TDD) entstehen Software-Einheiten und ihre Schnittstellen auf unterster Ebene, welche die darüberliegenden Software-Strukturen verfeinern.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.5	BC	ensure that SOFTWARE UNITS meet acceptance criteria	Durch Test-getriebene Entwicklung (TTD) wird sichergestellt, dass die entwickelte Software entsprechende Akzeptanzkriterien auf unterster Ebene befriedigt.
§ 5.5	ABC	write and verify the code for the SOFTWARE UNITS. The detailed design is to be translated into source code.	Mit Test-Driven Development (TDD) entsteht das Design des zu erstellenden Codes durch ausführbare Schnittstellen-Tests. Diese werden dann durch produktiven Code erfolgreich durchlaufen.
§ 5.5	BC	establish strategies, methods and procedures for verifying each SOFTWARE UNIT. Where VERIFICATION is done by testing, the test procedures shall be EVALUATED for correctness.	Durch Test-getriebene Entwicklung wird die Software-Verifizierung sichergestellt. TDD ist ein definierter, korrekter Prozess.

Verknüpfung mit anderen Praktiken und Artefakten:

- **Architecture Documentation** - Die Architektur Dokumentation auf Detailebene entspricht der Menge aller ausführbaren Tests, welche durch Test-Driven Development entstehen.

Unit Testing

Beschreibung

Unit Tests sind kleine Programme, welche automatisiert gegen Schnittstellen von Methoden auf unterster Code-Ebene testen.

Unit Tests werden idealerweise durch die Anwendung von Test-Driven Development (TDD) erzeugt und damit mit dem Test-First-Verfahren entwickelt: zunächst wird ein Unit-Test implementiert, welcher eine noch nicht existierende Funktionalität testet. Dieser Unit-Test schlägt fehl (wird rot). Die nicht existierende Funktionalität wird dann mit dem einfachsten Mittel realisiert, bis der Unit-Test durchläuft (grün wird). Danach kann der Code der umgesetzten Funktionalität durch geeignetes Refactoring verbessert und „clean“ gemacht werden.

Unit-Tests werden programmiert und gehören damit genau so zum Code, wie jeder „normale“ Code auch. Dies wird oftmals vergessen mit dem Effekt, dass die Unit-Test-Code-Basis nach und nach erodiert und selbst in die Wartbarkeitsfalle hineinfließt. Durch regelmäßiges Refactoring des Test-Codes kann dieser Effekt vermieden werden.

Zum einen sind Unit-Tests White-Box-Tests gegen Schnittstellen auf tiefster Code-Ebene, zum anderen finden sich unter dem Begriff Unit-Test auch allgemeine Tests auf der Ebene beliebiger Software-Einheiten. Die Terminologie „Unit-Test“ führt immer wieder zu Verwirrungen. Eine team- oder organisations-spezifische Definition des Begriffes "Unit Test" ist daher im Vorfeld notwendig.

Verantwortliche/beteiligte Rollen

- Entwicklungs-Team

Referenzen

- http://en.wikipedia.org/wiki/Unit_testing
- <http://c2.com/cgi/wiki?UnitTest>
- <http://www.extremeprogramming.org/rules/unittests.html>
- <http://guide.agilealliance.org/guide/unittest.html>
- <http://artofunittesting.com/>
- Roy Osherove - The Art of Unit Testing

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.4	ABC	SOFTWARE UNITS can be tested separately	Software-Einheiten dürfen mit Unit Tests getestet werden, ohne die Einbindung anderer Software-Einheiten.
§ 5.5	BC	perform the SOFTWARE UNIT VERIFICATION and document the results	Die Software wird durch Unit Tests auf unterster Ebene verifiziert. Durch Test-Automation kann eine Dokumentation der Testergebnisse erzeugt werden.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.5	C	include additional acceptance criteria as appropriate for: proper event sequence; data and control flow; planned resource allocation; fault handling; initialisation of variables; self-diagnostics; memory management; memory overflows; boundary conditio	Technische Akzeptanzkriterien, wie z.B. Sequenzen, Abläufe, Kontrollflüsse, Fehlerbehandlung, Initialisierung und Grenzwerte, werden mit Hilfe von Unit Tests definiert.
§ 5.6	ABC	In order to fully test a SOFTWARE PRODUCT both black and white box testing might be required.	Unit Tests entsprechen Black-Box-Tests bezüglich der atomaren Schnittstellen und White-Box-Tests bezüglich der Software-Strukturen.

Usability Process

Beschreibung

Die Usability-Prozesse dienen dazu, das vom System bereitgestellte User Interface auf Gebrauchstauglichkeit hin zu untersuchen, zu prüfen, sowie entsprechendes Anwenderfeedback zu erhalten.

In agilen Methoden erhalten wir dieses Feedback durch die direkte Einbindung der echten Anwender unseres Systems. Dies kann beispielsweise im Sprint Review Meeting geschehen, aber auch in eigenen Usability-Veranstaltungen beim Anwender.

In diesem Buch wird nicht näher auf Usability-Prozesse eingegangen, da diese in einer eigenen Norm IEC 62366 behandelt und für viele Beispiele im Detail beschrieben sind.

Verantwortliche/beteiligte Rollen

- Product Owner
- Qualitätsmanager

Referenzen

- IEC 62366:2007 Medizinprodukte - Anwendung der Gebrauchstauglichkeit auf Medizinprodukte
- <http://guide.agilealliance.org/guide/usability.html>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.2	ABC	IEC 60601-1-6 Usability	Ein Gebrauchstauglichkeitsprozess ist definiert und wird umgesetzt.

Zero Bug Tolerance

Beschreibung

Zero Bug Tolerance sorgt dafür, dass während einer Produktentwicklung keine Fehler verwaltet werden, sondern dass diese behoben werden. Das Verwalten von Fehlern und die Bewertung und Auswertung dadurch entstehender Fehlerlisten kostet eine große Menge an Zeit und Energie, welche die Effizienz für wichtige Entwicklungsaufgaben reduziert.

Auf den ersten Blick mag es utopisch und unrealistisch klingen, keine Fehler verwalten zu wollen. Dies führt in letzter Konsequenz dazu, kein Fehler-Tracking-System mehr zu benötigen.

Die Anwendung der Zero Bug Tolerance ist jedoch denkbar einfach:

- Entdeckte/gefundene Fehler werden schnellstmöglich von Team und Product Owner bewertet. Sie verbleiben nicht erst lange „auf Halde“ in einem eigenen Verwaltungssystem, sondern kommen direkt zur Bewertung auf den Tisch.
- Entweder handelt es sich um einen dringlichen Fehler, welche eine bereits umgesetzte Funktionalität beeinträchtigt. Dann wird dieser Fehler weit oben im Product Backlog einsortiert, um möglichst in der nächsten Iteration behoben werden zu können.
- Oft handelt es sich um einen nicht nachvollziehbaren oder sogar unwichtigen „Fehler“, der die Funktionalität für den Anwender in keiner Weise einschränkt. Diese Art von Fehler

können direkt gelöscht werden¹⁴, da der Nutzen ihrer Behebung in keinem Verhältnis zum Aufwand steht.

- Manchmal wird aus einer Fehlermeldung ein Feature-Wunsch. In diesem Fall kann ein entsprechender Eintrag vom Product Owner im Product Backlog angelegt werden.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team
- Scrum Master

Referenzen

- <http://www.infoq.com/news/2011/02/zero-defect-systems>
- <http://www.joelonsoftware.com/articles/fog0000000043.html> (Punkt 5)
- <http://sdt.bz/32548>

¹⁴ Dieses Vorgehen erfordert Mut und die Einsicht, dass durch das Aufbewahren alter und unwichtiger Fehlermeldungen kein Mehrwert für eine spätere Fehlerbehebung gewonnen wird. Sollte Ihr Umfeld dieses Vorgehen nicht zulassen, dann versuchen Sie zumindest, solche Fehlermeldungen in ein Archiv zu verschieben, welches nicht jederzeit sichtbar ist und den Blick auf das Wesentliche verschleiert.

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.6	BC	enter ANOMALIES found during software integration and integration testing into a software problem resolution PROCESS	Eine konsequente Zero-Bug-Tolerance führt dazu, dass alle Fehler, welche während der Software-Integration gefunden werden, schnellstmöglich behoben oder als irrelevant definiert werden.
§ 5.7	BC	enter ANOMALIES found during software system testing into a software problem resolution PROCESS	Eine konsequente Zero-Bug-Tolerance führt dazu, dass alle Fehler, welche während der Software- und System-Tests gefunden werden, schnellstmöglich behoben oder als irrelevant definiert werden.
§ 5.7	ABC	If a decision has been made not to fix anomalies, they need to be EVALUATED in relation to the HAZARD analysis and the SAFETY of the device. Root cause and symptoms of ANOMALIES, and the rationale for not fixing them should be documented.	Bei einer Zero Bug Tolerance muss nicht jedes Problem und nicht jeder Fehler behoben werden. Falls entschieden wird, einen Fehler nicht zu beheben, muss die Begründung dokumentiert werden. Dies kann direkt im entsprechenden Backlog-Eintrag erfolgen.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.8	BC	EVALUATE known residual ANOMALIES to ensure that they do not contribute to an unacceptable RISK	Bei Anwendung der Zero Bug Tolerance durchläuft jeder Fehler und jede Abweichung die Risiko-Analyse. Nur wenn ein Fehler und eine Abweichung nicht zu einem inakzeptablen Risiko beitragen, darf entschieden werden, den Fehler und die Abweichung nicht zu beheben.

Dokumente, Artefakte, Pläne

Architecture Documentation

Beschreibung

Architektur-Dokumentation wird von der IEC 62304 gefordert und kann auf Detailebene durch die Menge aller ausführbaren Tests aus dem Test-Driven Development (TDD), auf höherer Ebene durch Architekturplanung im Sprint Planning Meeting beschrieben werden.

Bei der Architektur-Dokumentation gibt es drei Aspekte, welche beschrieben werden sollten:

- Design-Entscheidungen und ihre Begründungen
- angewendete Design Patterns
- eingesetzte Entwicklungs-Frameworks

Es ist nicht wichtig, jede einzelne Code-Zeile in die Architektur-Dokumentation einfließen zu lassen. Sie sollte aus Sicht des Lesers geschrieben sein, so dass sich neue Teammitglieder (Entwickler und Tester) leicht zurecht finden können. Dies bedeutet auch, dass sich die Architektur-Dokumentation auf möglichst wenige Seiten beschränkt.

Durch entsprechende Werkzeuge lässt sich Architektur-Dokumentation in UML-Form automatisch erzeugen, so dass jederzeit die aktuelle Architektur eingesehen werden kann.

Verantwortliche/beteiligte Rollen

- Entwicklungs-Team

Referenzen

- <http://www.oose.de/teamblog/team/frag-das-team-fur-architekturdocumentation/>
- <http://efasoft.blogspot.de/2010/10/software-architecture-documentation-in.html>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.3	BC	Document the architecture so that it can be verified.	Mit der Architektur-Dokumentation ist es möglich, die Architektur zu verifizieren.
§ 5.3	BC	transform the requirements for the MEDICAL DEVICE SOFTWARE into a documented ARCHITECTURE that describes the software's structure	Die umgesetzte Software-Architektur kann mit Hilfe von UML-Diagrammen beschrieben und dokumentiert werden.

Definition of Done

Beschreibung

Die Definition-of-Done beschreibt alle Aktivitäten bzw. Zustände, welche eine Anforderung durchlaufen muss, bevor sie als "abgeschlossen" betrachtet werden darf.

Ein typischer Dialog zwischen Projektleiter und Entwickler lautet wie folgt.

Projektleiter: „Ist das Feature fertig?“

Entwickler: „Ja, das Feature funktioniert und ist fertig.“

Projektleiter: „Also können wir es zum Kunden ausrollen?“

Entwickler: „Nein, dafür müssen die Tester nochmal ran und es muss noch ins Deployment aufgenommen werden.“

Während der kurzen Iterationen agiler Prozesse möchten wir sicher sein, dass eine Anforderung tatsächlich umgesetzt ist und wir diese nicht noch einmal anfassen müssen, um sie abzuschließen. Was dies tatsächlich bedeutet, wird über die Definition-of-Done festgelegt.

Die Definition-of-Done kann für jedes Team unterschiedlich sein, da es in jedem Umfeld eigene Aspekte gibt, die vom Team umgesetzt werden müssen. Wichtig ist in jedem Fall, dass zwischen Product Owner und Entwicklungs-Team Einigkeit darüber herrscht, was die Aussage „die Anforderung ist fertig“ bedeutet.

Die einzelnen Aspekte einer Definition-of-Done betrachten nicht die Funktionalität einer Anforderung, sondern deren Qualität.

Typische Inhalte einer Definition-of-Done lauten beispielsweise:

- Code ist im Versionierungssystem eingecheckt
- Build-Server läuft ohne Warnungen und Fehler durch
- alle Unit-Tests sind nach dem Build grün
- Pair Programming oder Code-Review wurde durchgeführt
- Product Owner hat „OK“ gegeben
- etc.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team
- Scrum Master

Referenzen

- <http://www.scrumalliance.org/articles/105-what-is-definition-of-done-dod>
- <http://www.scrum.org/Resources/Scrum-Glossary/Definition-of-Done>
- <http://guide.agilealliance.org/guide/sashimi.html>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 1	ABC	all PROCESS outputs should be available when needed as input to further work on the software	Mit der Einhaltung einer Definition-of-Done (DoD) wird sichergestellt, dass fertiggestellte Ergebnisse weiterverarbeitet werden können.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.1	ABC	include or reference in the software development plan, a plan to conduct the ACTIVITIES and TASKS of the software RISK MANAGEMENT PROCESS, including the management of RISKS relating to SOUP	Der Risiko-Management-Prozess ist für jede einzelne Anforderung in der Definition-of-Done enthalten.
§ 5.1	ABC	include or reference in the software development plan: the required VERIFICATION TASKS for each life cycle ACTIVITY	Die Definition-of-Done enthält alle Verifizierungsaufgaben, die für eine Anforderung durchgeführt werden müssen.
§ 5.1	ABC	establish a software development plan (or plans) for conducting the ACTIVITIES of the software development PROCESS appropriate to the scope, magnitude, and software safety classifications of the SOFTWARE SYSTEM to be developed	Die Definition-of-Done beschreibt alle durchzuführenden Aktivitäten, um eine Anforderung vollständig umzusetzen.
§ 5.1	BC	plan to place CONFIGURATION ITEMS under documented configuration management control before they are VERIFIED	Die Definition-of-Done beinhaltet das Einchecken von umgesetzten Anforderungen in ein Versionierungssystem.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.6	ABC	Software integration and system testing can be performed in a simulated environment, on actual target hardware, or on the full MEDICAL DEVICE.	Eine Anforderung oder User Story kann als erledigt betrachtet werden, wenn die entsprechenden Integrations- und System-Tests mit einer Simulation, einem Prototypen oder einem Mock durchgeführt wurden.
§ 5.7	BC	verify that the VERIFICATION strategies and the test procedures used are appropriate	Die Definition-of-Done stellt sicher, dass Verifikation und entsprechende Test-Prozeduren eingehalten werden.
§ 5.8	BC	ensure that software VERIFICATION has been completed and the results EVALUATED before the software is released	In der Definition-of-Done wird festgelegt, dass die Software bzw. jede User Story verifiziert sein muss, bevor sie ausgeliefert werden darf.
§ 5.8	BC	ensure that all ACTIVITIES and TASKS are complete along with all the associated documentation	Durch die Definition-of-Done wird sichergestellt, dass alle Aufgaben zum Zeitpunkt eines Software-Release abgeschlossen sind.

Definition of Ready

Beschreibung

Die Definition-of-Ready beschreibt alle Aktivitäten bzw. Zustände, welche eine Anforderung durchlaufen muss, bevor sie bereit ist, in einen Sprint aufgenommen und umgesetzt zu werden.

Wie die Definition-of-Done kann auch die Definition-of-Ready für jedes Team unterschiedlich sein.

Das wesentliche Ziel der Definition-of-Ready ist das gemeinsame Verständnis zwischen Product Owner und Entwicklungs-Team darüber, welche Idee eine Anforderung oder User Story repräsentiert und welcher Anwendernutzen tatsächlich umgesetzt werden soll.

Typische Aspekte einer Definition-of-Ready beinhalten beispielsweise:

- Eine Story ist geschätzt, was bedeutet, dass eine inhaltliche Diskussion zwischen Product Owner und Entwicklungs-Team stattgefunden hat.
- Eine Story ist klein genug, um innerhalb einer Iteration umgesetzt werden zu können.
- Eine Story hat definierte Akzeptanzkriterien, damit sie während der Iteration vom Team geprüft werden kann.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team
- Scrum Master

Referenzen

- <http://www.romanpichler.com/blog/product-backlog/the-definition-of-ready/>
- <http://guide.agilealliance.org/guide/ready.html>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.2	ABC	an evaluation of the requirements is required before the requirements analysis ACTIVITY is complete	Die Definition-of-Ready einer Anforderung beinhaltet eine Bewertung durch das Team darüber, ob diese Anforderung bereit ist, in einem Sprint umgesetzt zu werden.
§ 7.1	BC	consider potential causes including incorrect or incomplete specification of functionality	Die Definition-of-Ready einer Anforderung fordert deren korrekte und vollständige Definition.
§ 8.2	ABC	change CONFIGURATION ITEMS only in response to an approved CHANGE REQUEST	Die Umsetzung von Anforderungen, welche das bestehende Software-System verändern (in einem inkrementell-iterativen Prozess also alle Anforderungen), muss vom Product Owner beauftragt sein. Dies kann mit Hilfe der Definition-of-Ready sichergestellt werden.

Iteration Notes (Sprint Notizen)

Beschreibung

Iteration Notes (Sprint Notizen) beschreiben:

- was in einer Iteration passiert ist,
- wer beteiligt war,
- wann die Iteration stattfand,
- welche Anforderungen umgesetzt wurden und
- welche Fehler, Probleme und Risiken identifiziert wurden.

Iteration Notes (Sprint Notizen) dienen zum Einen der Stakeholder-Kommunikation mit Kunden, Anwendern und Management, die nicht am Sprint Review Meeting teilnehmen können. Zum Anderen wird auf diesem Wege eine Projektdokumentation aufgebaut, welche den Entstehungsprozess des Systems für Auditoren, aber auch für neue Teammitglieder nachvollziehbar macht.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team

Referenzen

- keine

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.6	BC	identify the tester	Die Iteration Notes (Sprint Notizen) beinhalten die Liste der in der Iteration beteiligten Team-Mitglieder, wie zum Beispiel auch die Tester.
§ 5.8	BC	document all known residual ANOMALIES	Nicht behobene Fehler und Abweichungen werden in den Iterations-Notizen festgehalten.
§ 7.3	BC	document the verification of implemented risk control measures	Die Iteration Notes (Sprint Notizen) beinhalten die umgesetzten Maßnahmen aus der Risikoanalyse und deren Verifizierung im aktuellen Sprint.
§ 9	ABC	maintain records of PROBLEM REPORTS and their resolution including their VERIFICATION	In den Iteration Notes (Sprint Notizen) sind alle Fehlerberichte aus externen Quellen enthalten, sowie deren abgeleitete Tests zur Verifikation.

Linkable IDs

Beschreibung

Verweisende IDs sind die Basis für sämtliche Anforderungen an die Traceability. Sie können automatisch z.B. über eine Versionskontrolle oder eine Anwendung zur Verwaltung des Product Backlogs vergeben werden. Auch die manuelle Vergabe bei der Arbeit mit einem reinen Papier-Product Backlog ist möglich.

In diesem Buch wird nicht weiter auf die Vergabe und den Umgang mit Linkable IDs eingegangen.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team

Referenzen

- keine

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.2	ABC	software requirements: are traceable to SYSTEM requirements or other source	User Stories und darüberliegende bzw. externe Anforderungen werden mit Hilfe eindeutiger, referenzierbarer IDs versehen.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.2	ABC	software requirements: can be uniquely identified	User Stories und darüberliegende bzw. externe Anforderungen werden mit Hilfe eindeutiger, referenzierbarer IDs versehen.
§ 7.3	BC	document TRACEABILITY of software HAZARDS	Mit Hilfe von verweisenden IDs kann die Traceability hergestellt werden von erkannten Gefahrensituationen über Maßnahmen über umzusetzende Software-Einheiten bis zur Verifikation dieser Software-Einheiten.

Product Backlog

Beschreibung

Das Product Backlog ist eine nach Wichtigkeit sortierte Liste aller Wünsche, Ideen und Anforderungen an das zu entwickelnde System. Das Product Backlog ist kein fixes Dokument mit starrem Umfang, sondern muss sich im Laufe der Entwicklung kontinuierlich verändern, um dem empirischen Prozess agiler Methoden genüge zu tun und Anwenderfeedback einfließen zu lassen.

Die wichtigsten, also obersten Einträge im Product Backlog werden mit Hilfe der Definition-of-Ready in einen Zustand gebracht, dass sie vom Entwicklungs-Team im Sprint Planning Meeting für die nächste Iteration eingeplant werden können.

Das Product Backlog hat nichts mit einer klassischen Software-Requirements-Specification (SRS) zu tun. Es ist ein lebendes Dokument, welches sich im Laufe der Zeit verändern soll, um die jeweiligen Erkenntnisse abzubilden. Das Product Backlog soll nur Einträge enthalten, die unmittelbar dazu beitragen, das Projekt- oder Produktziel zu erreichen.

Der Product Owner trägt die Verantwortung für das Product Backlog und dessen kontinuierlicher Pflege.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team

Referenzen

- <http://www.mountangoatsoftware.com/scrum/product-backlog>
- <http://guide.agilealliance.org/guide/backlog.html>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.2	ABC	include in the software requirements: user documentation to be developed	Zu erzeugende Anwender-Dokumentation wird als Anforderung im Product Backlog verwaltet.
§ 5.2	ABC	include in the software requirements: regulatory requirements	Anforderungen aus Normen und Richtlinien werden im Product Backlog verwaltet.
§ 5.2	ABC	include in the software requirements: SECURITY requirements	Das Product Backlog enthält alle Sicherheits-Anforderungen (Security).
§ 5.2	ABC	software requirements: implement SYSTEM requirements including those relating to RISK CONTROL	Das Product Backlog enthält alle Anforderungen, die sich aus dem Risikomanagement-Prozess ergeben.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.2	ABC	define and document SOFTWARE SYSTEM requirements from the SYSTEM level requirements	Im Product Backlog befinden sich grobe System-Anforderungen, welche iterativ in kleinere Software-Anforderungen verfeinert werden.
§ 5.7	BC	enter ANOMALIES found during software system testing into a software problem resolution PROCESS	Fehler, welche während der System-Verifikation gefunden und nicht sofort behoben werden, werden in das Product Backlog einsortiert.
§ 5.8	BC	ensure that the released SOFTWARE PRODUCT can be reliably delivered: replication, media labelling, packaging, protection, storage, and delivery.	Im Product Backlog sind Maßnahmen und Aktivitäten enthalten, die eine verlässliche Auslieferung der Software gewährleisten.

Release Notes

Beschreibung

Release Notes beinhalten alle für den Anwender, Kunden und Operator des Systems notwendigen Informationen, um das System aufsetzen und benutzen zu können.

Dieses Buch geht nicht weiter auf das Thema „Release Notes“ ein, da es sich nicht um ein spezifisches, agiles Artefakt handelt.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team

Referenzen

- keine

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.8	ABC	document the VERSION of the SOFTWARE PRODUCT that is being released	In den Release Notes ist die Version des Software-Release enthalten.

§	SK	Normtext	Herstellung der Normkonformität
§ 6.2	ABC	inform users and regulators about: the nature of any available changes to released SOFTWARE PRODUCTS and how to obtain and install the changes	Die Release Notes einer Software beinhalten alle Änderungen an dem Software-System seit dem letzten Release.

Software Development Plan

Beschreibung

Der Software-Entwicklungsplan beschreibt die grundlegenden Vorgehensweisen für die Produktentwicklung, sowie die eingesetzten Werkzeuge und die Teamzusammensetzung.

Dieses Buch geht nicht weiter auf das Thema „Software Development Plan“ ein, da es sich dabei nicht um ein spezifisches, agiles Artefakt handelt.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team
- Scrum Master
- Qualitätsmanager

Referenzen

- IEC 62304

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.1	ABC	include or reference in the software development plan procedures for coordinating the software development and the design and development validation necessary to satisfy 4.1 (quality management system)	Der Software-Entwicklungsplan erwähnt Scrum als eingesetztes, agiles Vorgehensmodell mit dem darin enthaltenen Sprint Review Meeting, in welchem die Forderungen aus Paragraph 4.1 umgesetzt werden: Sicherstellung der Umsetzung von Kundenanforderungen sowie regulatorischen Anforderungen.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.1	ABC	<p>reference in the software development plan information about the documents to be produced:</p> <p>title</p> <p>purpose</p> <p>intended audience of document</p> <p>procedures and responsibilities for development, review, approval and modification</p>	<p>Im Software Development Plan werden folgende Scrum-Artefakte aufgeführt:</p> <ul style="list-style-type: none"> * Product Backlog * Sprint Backlog * Definition of Done * Team Charter / Working Agreements * Definition of Ready * Iteration Notes (Sprint Notizen) * Architecture Documentation
§ 5.8	BC	<p>document the procedure and environment used to create the released software</p>	<p>Der Software-Entwicklungsplan beschreibt die Vorgehensweise zur Erstellung eines Software-Release, sowie das dafür notwendige Environment.</p>

Software Maintenance Plan

Beschreibung

Der Software-Wartungsplan beschreibt die Prozesse zur Fehlerbehandlung und dem Umgang mit Anwenderfeedback nach der Markteinführung eines Produktes.

Dieses Buch geht nicht weiter auf das Thema „Software Maintenance Plan“ ein, da es sich dabei nicht um ein spezifisches, agiles Artefakt handelt.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team
- Qualitätsmanager

Referenzen

- keine

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 6.1	ABC	software maintenance plan includes criteria for determining whether feedback is considered to be a problem	Im Software Maintenance Plan wird festgelegt, anhand welcher Kriterien ein Feedback als Problem/Fehler betrachtet wird.

§	SK	Normtext	Herstellung der Normkonformität
§ 6.1	ABC	software maintenance plan includes use of the software configuration management PROCESS for managing modifications to the existing SYSTEM	Im Software Maintenance Plan wird der Einsatz des Versionierung-Systems beschrieben.
§ 6.1	ABC	establish a software maintenance plan for conducting the ACTIVITIES and TASKS of the maintenance PROCESS	Im Software Maintenance Plan werden alle Aktivitäten und Aufgaben des Maintenance Prozesses beschrieben.
§ 6.1	ABC	software maintenance plan includes procedures for: receiving, documenting, evaluating, resolving and tracking feedback arising after release of the MEDICAL DEVICE SOFTWARE	Im Software Maintenance Plan wird festgelegt, wie nach einem Release das Feedback eingeholt, dokumentiert, ausgewertet, umgesetzt und verfolgt wird.

Sprint Development Plan = Sprint Backlog

Beschreibung

Das Sprint Backlog ist das Ergebnis des Sprint Planning Meetings und beinhaltet alle Anforderungen und Aktivitäten, die während des nächsten Sprints umgesetzt werden sollen.

Im Sprint Planning Meeting nimmt sich das Entwicklungs-Team so viele Einträge aus dem priorisierten Product Backlog, wie es zutraut, im nächsten Sprint umsetzen zu können. Diese Einträge repräsentieren das Sprint Backlog. Im zweiten Teil des Sprint Planning Meetings erzeugt das Entwicklungs-Team für jeden einzelnen Eintrag im Sprint Backlog die entsprechenden Tasks, die durchgeführt werden müssen, um den Eintrag umzusetzen.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team
- Scrum Master

Referenzen

- <http://www.mountangoatsoftware.com/scrum/sprint-backlog>
- <http://guide.agilealliance.org/guide/iteration.html>
- <http://www.scrumalliance.org/articles/122--tips-for-creating-a-good-sprint-backlog>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.1	BC	include or reference in the software development plan, a plan to integrate the SOFTWARE ITEMS (including SOUP) and perform testing during integration.	Das Sprint Backlog enthält alle Anforderungen, welche innerhalb eines Sprints in das Gesamtsystem integriert und integrationsgetestet werden.
§ 5.1	ABC	include or reference in the software development plan: DELIVERABLES requiring VERIFICATION	Im Sprint Backlog sind alle Produktergebnisse enthalten, welche auch im Sprint verifiziert werden müssen.
§ 5.1	ABC	As inputs for software development, SYSTEM requirements shall be referenced in the software development plan	Das Sprint Backlog beinhaltet iterativ-inkrementell auch alle Anforderungen, welche sich aus dem Gesamtsystem an die Software ergeben.
§ 5.1	ABC	update the software development plan as development proceeds as appropriate	Der Software-Entwicklungsplan wird iterativ-inkrementell verändert durch die entstehenden Sprint Backlogs.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.6	BC	enter ANOMALIES found during software integration and integration testing into a software problem resolution PROCESS	Fehler, welche während der Software-Integration auftauchen, werden im Sprint Backlog gepflegt, um sie schnellstmöglich zu beheben.
§ 9	ABC	maintain records of PROBLEM REPORTS and their resolution including their VERIFICATION	Fehlerberichte werden im Sprint Backlog verwaltet, um schnellstmöglich behoben zu werden.

Team Charter / Working Agreements

Beschreibung

Ein Team Charter beschreibt die grundlegenden Arbeitsweisen innerhalb eines Teams, die eingesetzten Methoden und Werkzeuge, sowie die Ziele und Vision des Projektes. Im weitesten Sinne gehören auch die Definition-of-Done und die Definition-of-Ready zu den Working Agreements.

Team Charter und Working Agreements unterscheiden sich von Team zu Team, da die Inhalte jeweils aus der vorhandenen Team-Kultur entstehen. Ob Pünktlichkeit auf der Liste steht oder eine Meetingkultur ohne Mobiltelefone verabschiedet wird, liegt ausschließlich am Team.

In diesem Buch wird das Thema nicht weiter behandelt, da es keine „Best Practice“-Beispiele gibt. Jedes Team muss für sich selbst definieren, wie es zusammen arbeiten möchte und welche teaminternen Regeln auf die Team Charter und Working Agreements geschrieben werden sollen.

Als Anhaltspunkt mögen folgende Fragen dienen:

- Welche Werte sind uns wichtig für die Zusammenarbeit als Team?
- Welche Verhaltensregeln wollen wir uns geben?
- Welche fachlichen Regeln an unsere Arbeit wollen wir uns geben?

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team

- Scrum Master

Referenzen

- <http://www.infoq.com/articles/agile-development-team-character>
- <http://agile.dzone.com/news/norms-values-working>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.1	BC	The items to be controlled shall include tools, items or settings, used to develop the MEDICAL DEVICE SOFTWARE, which could impact the MEDICAL DEVICE SOFTWARE.	Die Working Agreements eines Teams beschreiben alle Tools und Dinge, welche das endgültige Produkt beeinflussen könnten.
§ 5.1	ABC	include or reference software configuration management information in the software development plan	Die Working Agreements eines Teams beschreiben das eingesetzte Versionierungssystem sowie dessen grundsätzliche Handhabung.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.1	C	include or reference in the software development plan: a) standards, b) methods, and c) tools associated with the development of SOFTWARE ITEMS of class C	In den Working Agreements eines Teams wird festgelegt, mit welchen Standards, Methoden und Tools die Umsetzung von Anforderungen in funktionierende Software geschieht.
§ 5.1	ABC	include or reference in the software development plan: milestones at which the DELIVERABLES are VERIFIED	Im Team Charter wird die Sprint-Länge festgelegt und damit die iterativen Meilensteine, zu denen umgesetzte Anforderungen verifiziert werden.

Test Documentation

Beschreibung

Test-Dokumentation soll nach Möglichkeit auf Knopfdruck erzeugt werden können und sich auf die wesentlichen Ergebnisse und Aussagen beschränken.

Die Test-Dokumentation beinhaltet folgende Fragmente:

- Software Development Plan: beschreibt die Test-Tool-Chain, sowie eine Liste der Tester (und Entwickler) im Team
- Sprint-Notizen: beschreiben gefundene und offene Fehler und Abweichungen, sowie die jeweils aktuelle Software-Version
- Ergebnisse der automatisierten Unit-, Integrations- und Regressionstests

Verantwortliche/beteiligte Rollen

- Entwicklungs-Team

Referenzen

- keine

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 9	ABC	include in the test documentation: test results; ANOMALIES found; VERSION of software tested; relevant hardware and software test configurations; relevant test tools; date tested; identification of the tester	In der Test-Dokumentation werden folgende Informationen abgelegt: Test-Ergebnisse, gefundene Abweichungen, getestete Software-Version, Hardware- und Software-Konfigurationen, eingesetzte Test-Tools, Test-Datum, Test-Personal

Verknüpfung mit anderen Praktiken und Artefakten:

- **Exploratory Testing** - Die benötigte Test Dokumentation wird mit den Ergebnissen der explorativen und manuellen Tests angereichert.

User Story

Beschreibung

Eine User Story steht in diesem Kontext für sämtliche Typen von Anforderungen, die in einem Product Backlog oder Sprint Backlog enthalten sein können.

Unter den Begriff fallen:

- User Story,
- technische Story,
- Fehlerbeschreibung,
- Problembeschreibung,
- Anforderung,
- etc.

Echte User Stories helfen dem Product Owner und dem Entwicklungs-Team dabei, den Fokus auf den Anwendernutzen zu legen. Bei jeder Diskussion über Inhalte und Anforderungen an das zu erstellende System sollen Fragen folgender Art gestellt werden:

- „Für welchen Anwender erzeugt dies genau welchen Nutzen?“
- „Welches Problem wird durch die Umsetzung dieser Story behoben?“

User Stories befinden sich anfänglich im Product Backlog und durchlaufen ihren Lebenszyklus über die Definition-of-Ready in das Sprint Backlog des Entwicklungs-Teams. Dort werden sie zerlegt in Tasks, die während der Iteration abgearbeitet werden, bis die User Story über die Definition-of-Done vom Product Owner abgenommen wird und aus der weiteren Betrachtung herausfällt.

Verantwortliche/beteiligte Rollen

- Product Owner
- Entwicklungs-Team

Referenzen

- Mike Cohn - User Stories Applied
- <http://guide.agilealliance.org/guide/stories.html>
- <http://www.extremeprogramming.org/rules/userstories.html>
- <http://www.mountaingoatsoftware.com/topics/user-stories>

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 4	ABC	when a SOFTWARE ITEM is decomposed into further SOFTWARE ITEMS, such SOFTWARE ITEMS shall inherit the software safety classification of the original SOFTWARE ITEM	User Stories, welche durch Story Splitting aus einer übergeordneten User Story entstanden sind, erben die Sicherheitsklassifizierung dieser übergeordneten User Story.
§ 4	ABC	document the software safety class assigned to each SOFTWARE SYSTEM in the RISK MANAGEMENT FILE	Die Sicherheitsklassifizierung wird als Attribut jeder User Story zugeordnet, welche die Risiko-Analyse durchlaufen hat.

§	SK	Normtext	Herstellung der Normkonformität
§ 4	ABC	For each SOFTWARE SYSTEM, until a software safety class is assigned, Class C requirements shall apply	Einer User Story, der noch keine Sicherheitsklassifizierung zugeordnet wurde, wird per Default die Sicherheitsklassifizierung C zugeordnet.
§ 5.1	ABC	include or reference in the software development plan: the acceptance criteria for VERIFICATION of the DELIVERABLES	Eine User Story besteht neben der reinen Beschreibung auch aus definierten Akzeptanztests, welche zur Verifizierung der Umsetzung benötigt werden.
§ 5.2	ABC	include in the software requirements: usability engineering requirements that are sensitive to human errors and training	Anforderungen an die Gebrauchstauglichkeit werden über User Stories definiert.
§ 5.2	ABC	include in the software requirements: user maintenance requirements	Anforderungen an die Wartung durch den Anwender werden mit User Stories definiert.
§ 5.2	ABC	include in the software requirements: data definition and database requirements	Daten-Definitionen und Datenbank-Anforderungen werden über entsprechende User Stories spezifiziert.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.2	ABC	include in the software requirements: SOFTWARE SYSTEM inputs and outputs	Die Eingaben und Ausgaben des gesamten Software-Systems werden über entsprechende User Stories definiert.
§ 5.2	ABC	include in the software requirements: installation and acceptance requirements of the delivered MEDICAL DEVICE SOFTWARE at the operation and maintenance site or sites	Anforderungen an die Installation des Software-Produktes und deren Akzeptanzkriterien werden über entsprechende User Stories beschrieben.
§ 5.2	ABC	include in the software requirements: interfaces between the SOFTWARE SYSTEM and other SYSTEMS	Schnittstellen von und zu anderen Systemen werden über User Stories beschrieben.
§ 5.2	ABC	include in the software requirements: functional and capability requirements	Eine User Story beschreibt eine funktionale Anforderung bzw. eine benötigte Fähigkeit des zu erstellenden Produkts.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.2	ABC	Requirements are to be identified in the software requirements in such a way as to make it possible to trace the RISK CONTROL measures to the software requirements.	Eine User Story beinhaltet alle ermittelten Maßnahmen aus ihrer eigenen Risiko-Analyse.
§ 5.2	ABC	include in the software requirements: software-driven alarms, warnings, and operator messages	Alarmer, Warnungen und andere wichtige Meldungen für den Anwender werden mit Hilfe von User Stories definiert.
§ 5.2	ABC	include in the software requirements: requirements related to methods of operation and maintenance	Anforderungen an Betrieb und Wartung des Systems werden mit User Stories beschrieben
§ 5.2	ABC	software requirements: are stated in terms that permit establishment of test criteria and performance of tests to determine whether the test criteria have been met	User Stories sind so formuliert, dass sich Test- und Akzeptanz-Kriterien ableiten lassen.

§	SK	Normtext	Herstellung der Normkonformität
§ 5.7	BC	establish and perform a set of tests, expressed as input stimuli, expected outcomes, pass/fail criteria and procedures, for conducting SOFTWARE SYSTEM testing, such that all software requirements are covered	Eine User Story beinhaltet alle Testfälle, welche notwendig sind, um die Anforderungen dieser User Story prüfen und sicherstellen zu können.
§ 5.7	BC	verify that SOFTWARE SYSTEM test procedures trace to software requirements	User Stories beinhalten neben der Definition einer Anforderung auch deren Akzeptanz- und Testkriterien.
§ 6.2	ABC	record PROBLEM REPORTS that include actual or potential adverse events, and deviations from specifications	User Stories können auch Fehler- und Problemberichte darstellen, welche Abweichungen der Anforderungen und der Spezifikation beinhalten.
§ 8.2	ABC	provide traceability of change requests, their approvals, and relevant problem reports	Eine User Story, welche auf Basis eines Problems formuliert wird, beinhaltet den zugehörigen Problembericht, um Traceability herzustellen.

§	SK	Normtext	Herstellung der Normkonformität
§ 9	ABC	for each problem detected prepare a PROBLEM REPORT classified as follows: type, scope, criticality.	In User Stories können Fehler- und Problemberichte beschrieben sein.
§ 9	ABC	create a CHANGE REQUEST(S) for actions needed to correct the problem, or document the rationale for taking no action	User Stories können Fehlermeldungen, Change Requests, sowie andere Einträge enthalten, um das Systemverhalten zu ändern oder zu korrigieren.
§ 9	ABC	document the outcome of the investigation and evaluation	Die Ergebnisse von Fehleranalysen und -bewertungen werden in der User Story bzw. dem Fehlereintrag direkt festgehalten.

Versioning System

Beschreibung

Ein Versionierungssystem ist ein grundlegendes Werkzeug für die Entwicklung von Softwareprodukten.

In diesem Buch wird das Thema „Versionierungssystem“ nicht näher beschrieben, da es sich nicht um ein spezifisches, agiles Artefakt handelt.

Verantwortliche/beteiligte Rollen

- Entwicklungs-Team

Referenzen

- keine

Umgesetzte Anforderungen aus der Norm

§	SK	Normtext	Herstellung der Normkonformität
§ 5.8	BC	archive the SOFTWARE PRODUCT, the CONFIGURATION ITEMS, and the documentation	Durch ein Versionierungssystem werden alle Bestandteile eines Software-Release archiviert.

§	SK	Normtext	Herstellung der Normkonformität
§ 8.1	ABC	document the set of CONFIGURATION ITEMS and their VERSIONS that comprise the SOFTWARE SYSTEM configuration	Alle zu einem Software-System gehörenden Bestandteile und deren Versionen werden in einem Versionierungssystem abgelegt. Aus diesem Versionierungssystem kann die gültige, vollständige System-Konfiguration aller Bestandteile festgestellt werden.
§ 8.1	ABC	establish a scheme for the unique identification of CONFIGURATION ITEMS and their VERSIONS	Mit einem Versionierungssystem können alle zu einem Software-System gehörenden Komponenten und deren Versionen identifiziert und verwaltet werden.
§ 8.1	ABC	document the title, the MANUFACTURER, and the unique SOUP designator of each SOUP CONFIGURATION ITEM being used	Die im Software-System eingesetzten Drittparty-Komponenten werden in ihrem Ursprungszustand mit einem Versionierungssystem verwaltet.
§ 8.3	ABC	retain retrievable records of the history of controlled CONFIGURATION ITEMS including SYSTEM configuration	Mit Hilfe des Versionierungssystems kann die Historie aller Bestandteile des Software-Systems nachvollzogen werden.

Verknüpfung mit anderen Praktiken und Artefakten:

- **Linkable IDs** - Verweisende Identifier (Linkable IDs) werden beim Einsatz eines Versionierungssystems automatisch vergeben, da Change Sets mit einheitlichem Zeitstempel versehen sind und komplette Versionierungsstände mit Labels versehen werden können.
- **Kontinuierliche Integration** - Kontinuierliche Integration bedient sich eines Versionierungssystems, um jederzeit kleine Änderungen am System integrieren zu können.
- **Early Integration** - Frühe Integration bedient sich eines Versionierungssystems, um Änderungen am System sofort integrieren zu können.

Von den Anforderungen der Norm zu Praktiken und Artefakten

Dieses Kapitel beinhaltet die Gesamttabelle aller Normtexte mit Verweis aus den entsprechenden Paragraphen und die Sicherheitsklassifizierung, sowie eine Beschreibung der jeweiligen Normkonformität und eine Praktik bzw. ein Artefakt, mit dem die Konformität hergestellt werden kann.

Hinweis! In der gekürzten Ausgabe dieses Buches sind die Inhalte dieses Kapitels nicht enthalten. Bei Interesse wenden Sie sich bitte an den Autor.

§ 1 - Scope

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	all PROCESS outputs should be available when needed as input to further work on the software	Mit der Einhaltung einer Definition-of-Done (DoD) wird sichergestellt, dass fertiggestellte Ergebnisse weiterverarbeitet werden können.	Definition of Done
ABC	All PROCESS outputs should be maintained in a consistent state	Im Sprint Review wird die Konsistenz geprüft zwischen der erzeugten Software, ihrer Anforderungen und weiterer Ergebnisse bzw. Inputs.	Sprint Review
ABC	Any waterfall, incremental, and evolutionary life cycle may be used.	Scrum ist ein gültiges Prozess-Framework für den Software-Life-Cycle.	Scrum

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	before any MEDICAL DEVICE SOFTWARE is released, all PROCESS outputs should be consistent with each other and all dependencies between PROCESS outputs explicitly or implicitly required by this standard should be observed	Im Sprint Review werden alle Ergebnisse des gesamten Software-Life-Cycles auf gegenseitige Konsistenz geprüft.	Sprint Review
ABC	perform MEDICAL DEVICE RISK MANAGEMENT that is compliant with ISO 14971	Risiko-Management wird in erster Linie im Sprint Planning durchgeführt.	Risk Analysis and Management

§ 4 - General requirements

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	assign to each SOFTWARE SYSTEM a software safety class (A, B, or C) according to the possible effects on the patient, operator, or other people resulting from a HAZARD to which the SOFTWARE SYSTEM can contribute	Im Sprint Planning wird jeder Anforderung und User Story (auf jeder Ebene) eine Sicherheitsklassifizierung zugeordnet durch die Anwendung der Risiko-Analyse.	Risk Analysis and Management
ABC	demonstrate the ability to provide MEDICAL DEVICE SOFTWARE that consistently meets applicable regulatory requirements	Im Sprint Review Meeting wird festgestellt, ob die Entwicklung des erzeugten Software-Produkts den normativen Anforderungen entspricht.	Sprint Review
ABC	demonstrate the ability to provide MEDICAL DEVICE SOFTWARE that consistently meets customer requirements	Im Sprint Review Meeting wird festgestellt, ob das erzeugte Software-Produkt die Bedürfnisse und Anforderungen der Anwender befriedigt.	Sprint Review

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	document the software safety class assigned to each SOFTWARE SYSTEM in the RISK MANAGEMENT FILE	Die Sicherheitsklassifizierung wird als Attribut jeder User Story zugeordnet, welche die Risiko-Analyse durchlaufen hat.	User Story
ABC	For each SOFTWARE SYSTEM, until a software safety class is assigned, Class C requirements shall apply	Einer User Story, der noch keine Sicherheitsklassifizierung zugeordnet wurde, wird per Default die Sicherheitsklassifizierung C zugeordnet.	User Story
ABC	RISK MANAGEMENT should be applied as an integral part of the development PROCESS	Risiko-Management wird im Sprint Planning Meeting durchgeführt.	Risk Analysis and Management
ABC	software engineering	Modernes Software-Engineering wird umgesetzt mit der Anwendung von Clean Code und SOLID-Prinzipien, TDD, Unit Tests, Design Patterns und vielen anderen Methoden, die hier nicht vollständig aufgeführt werden können.	Clean Code / SOLID Principles

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	<p>The software ARCHITECTURE should promote segregation of software items that are required for safe operation and should describe the methods used to ensure effective segregation of those SOFTWARE ITEMS.</p>	<p>Durch die korrekte Anwendung objektorientierter Prinzipien wie "Separation of Concerns" und "Single Responsibility Principle" entstehen lose gekoppelte Architekturen.</p>	<p>Clean Code / SOLID Principles</p>
ABC	<p>use of a quality management system that complies with:</p> <p>ISO 13485; or</p> <p>a national quality management system standard; or</p> <p>a quality management system required by national regulation</p>	<p>Ein Qualitäts-Management-Prozess ist definiert und wird eingesetzt.</p>	<p>Quality Management Prozess</p>

SK	Normtext	Herstellung der Normkonformität	Praktik/Artefakt
ABC	when a SOFTWARE ITEM is decomposed into further SOFTWARE ITEMS, such SOFTWARE ITEMS shall inherit the software safety classification of the original SOFTWARE ITEM	User Stories, welche durch Story Splitting aus einer übergeordneten User Story entstanden sind, erben die Sicherheitsklassifizierung dieser übergeordneten User Story.	User Story

§ 5 - Software development process

§ 5.1 - Software development planning

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	As inputs for software development, SYSTEM requirements shall be referenced in the software development plan	Das Sprint Backlog beinhaltet iterativ-inkrementell auch alle Anforderungen, welche sich aus dem Gesamtsystem an die Software ergeben.	Sprint Development Plan = Sprint Backlog
ABC	establish a software development plan (or plans) for conducting the ACTIVITIES of the software development PROCESS appropriate to the scope, magnitude, and software safety classifications of the SOFTWARE SYSTEM to be developed	Die Definition-of-Done beschreibt alle durchzuführenden Aktivitäten, um eine Anforderung vollständig umzusetzen.	Definition of Done

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	include or reference in the software development plan procedures for coordinating the software development and the design and development validation necessary to satisfy 4.1 (quality management system)	Der Software-Entwicklungsplan erwähnt Scrum als eingesetztes, agiles Vorgehensmodell mit dem darin enthaltenen Sprint Review Meeting, in welchem die Forderungen aus Paragraph 4.1 umgesetzt werden: Sicherstellung der Umsetzung von Kundenanforderungen sowie regulatorischen Anforderungen.	Software Development Plan
ABC	include or reference in the software development plan, a plan to conduct the ACTIVITIES and TASKS of the software RISK MANAGEMENT PROCESS, including the management of RISKS relating to SOUP	Der Risiko-Management-Prozess ist für jede einzelne Anforderung in der Definition-of-Done enthalten.	Definition of Done

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	include or reference in the software development plan, a plan to integrate the SOFTWARE ITEMS (including SOUP) and perform testing during integration.	Konsequent angewandte Continuous Integration ist die Aktivität, durch welche eine frühzeitige Integration aller beteiligten Komponenten inklusive der dazugehörigen Integrationstests sichergestellt wird.	Continuous Integration
BC	include or reference in the software development plan, a plan to integrate the SOFTWARE ITEMS (including SOUP) and perform testing during integration.	Das Sprint Backlog enthält alle Anforderungen, welche innerhalb eines Sprints in das Gesamtsystem integriert und integrationsgetestet werden.	Sprint Development Plan = Sprint Backlog
C	include or reference in the software development plan: a) standards, b) methods, and c) tools associated with the development of SOFTWARE ITEMS of class C	In den Working Agreements eines Teams wird festgelegt, mit welchen Standards, Methoden und Tools die Umsetzung von Anforderungen in funktionierende Software geschieht.	Team Charter / Working Agreements

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	include or reference in the software development plan: DELIVERABLES requiring VERIFICATION	Im Sprint Backlog sind alle Produktergebnisse enthalten, welche auch im Sprint verifiziert werden müssen.	Sprint Development Plan = Sprint Backlog
ABC	include or reference in the software development plan: milestones at which the DELIVERABLES are VERIFIED	Im Team Charter wird die Sprint-Länge festgelegt und damit die iterativen Meilensteine, zu denen umgesetzte Anforderungen verifiziert werden.	Team Charter / Working Agreements
ABC	include or reference in the software development plan: the acceptance criteria for VERIFICATION of the DELIVERABLES	Eine User Story besteht neben der reinen Beschreibung auch aus definierten Akzeptanztests, welche zur Verifizierung der Umsetzung benötigt werden.	User Story
ABC	include or reference in the software development plan: the required VERIFICATION TASKS for each life cycle ACTIVITY	Die Definition-of-Done enthält alle Verifizierungsaufgaben, die für eine Anforderung durchgeführt werden müssen.	Definition of Done

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	include or reference software configuration management information in the software development plan	Die Working Agreements eines Teams beschreiben das eingesetzte Versionierungssystem sowie dessen grundsätzliche Handhabung.	Team Charter / Working Agreements
BC	plan to place CONFIGURATION ITEMS under documented configuration management control before they are VERIFIED	Die Definition-of-Done beinhaltet das Einchecken von umgesetzten Anforderungen in ein Versionierungssystem.	Definition of Done

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	<p>reference in the software development plan information about the documents to be produced:</p> <p>title</p> <p>purpose</p> <p>intended audience of document</p> <p>procedures and responsibilities for development, review, approval and modification</p>	<p>Im Software Development Plan werden folgende Scrum-Artefakte aufgeführt:</p> <ul style="list-style-type: none"> * Product Backlog * Sprint Backlog * Definition of Done * Team Charter / Working Agreements * Definition of Ready * Iteration Notes (Sprint Notizen) * Architecture Documentation 	<p>Software Development Plan</p>
BC	<p>The items to be controlled shall include tools, items or settings, used to develop the MEDICAL DEVICE SOFTWARE, which could impact the MEDICAL DEVICE SOFTWARE.</p>	<p>Die Working Agreements eines Teams beschreiben alle Tools und Dinge, welche das endgültige Produkt beeinflussen könnten.</p>	<p>Team Charter / Working Agreements</p>

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	update the software development plan as development proceeds as appropriate	Der Software-Entwicklungsplan wird iterativ-inkrementell verändert durch die entstehenden Sprint Backlogs.	Sprint Development Plan = Sprint Backlog

§ 5.2 - Software requirements analysis

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	an evaluation of the requirements is required before the requirements analysis ACTIVITY is complete	Die Definition-of-Ready einer Anforderung beinhaltet eine Bewertung durch das Team darüber, ob diese Anforderung bereit ist, in einem Sprint umgesetzt zu werden.	Definition of Ready
ABC	define and document SOFTWARE SYSTEM requirements from the SYSTEM level requirements	Im Product Backlog befinden sich grobe System-Anforderungen, welche iterativ in kleinere Software-Anforderungen verfeinert werden.	Product Backlog
ABC	ensure that existing requirements, including SYSTEM requirements, are re-EVALUATED and updated as appropriate as a result of the software requirements analysis ACTIVITY	Im Product Backlog Grooming Meeting werden bestehende Anforderungen und User Stories im Product Backlog neu bewertet und aktualisiert.	Product Backlog Grooming

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	ensure that existing requirements, including SYSTEM requirements, are re-EVALUATED and updated as appropriate as a result of the software requirements analysis ACTIVITY	Im Sprint Planning Meeting werden bestehende Anforderungen und User Stories im Product Backlog neu bewertet und aktualisiert.	Sprint Planning
ABC	ensure that existing requirements, including SYSTEM requirements, are re-EVALUATED and updated as appropriate as a result of the software requirements analysis ACTIVITY	Die Ergebnisse und Erkenntnisse aus dem Sprint Review Meeting fließen als Feedback direkt in die Anforderungen ein und können im anschließenden Sprint Planning Meeting bedacht werden.	Sprint Review
ABC	IEC 60601-1-6 Usability	Ein Gebrauchstauglichkeitsprozess ist definiert und wird umgesetzt.	Usability Process
ABC	include in the software requirements: data definition and database requirements	Daten-Definitionen und Datenbank-Anforderungen werden über entsprechende User Stories spezifiziert.	User Story

SK	Normtext	Herstellung der Normkonformität	Praktik/Artefakt
ABC	include in the software requirements: functional and capability requirements	Eine User Story beschreibt eine funktionale Anforderung bzw. eine benötigte Fähigkeit des zu erstellenden Produkts.	User Story
ABC	include in the software requirements: installation and acceptance requirements of the delivered MEDICAL DEVICE SOFTWARE at the operation and maintenance site or sites	Anforderungen an die Installation des Software-Produktes und deren Akzeptanzkriterien werden über entsprechende User Stories beschrieben.	User Story
ABC	include in the software requirements: interfaces between the SOFTWARE SYSTEM and other SYSTEMS	Schnittstellen von und zu anderen Systemen werden über User Stories beschrieben.	User Story
ABC	include in the software requirements: regulatory requirements	Anforderungen aus Normen und Richtlinien werden im Product Backlog verwaltet.	Product Backlog

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	include in the software requirements: requirements related to methods of operation and maintenance	Anforderungen an Betrieb und Wartung des Systems werden mit User Stories beschrieben	User Story
ABC	include in the software requirements: SECURITY requirements	Das Product Backlog enthält alle Sicherheits-Anforderungen (Security).	Product Backlog
ABC	include in the software requirements: SOFTWARE SYSTEM inputs and outputs	Die Eingaben und Ausgaben des gesamten Software-Systems werden über entsprechende User Stories definiert.	User Story
ABC	include in the software requirements: software-driven alarms, warnings, and operator messages	Alarmer, Warnungen und andere wichtige Meldungen für den Anwender werden mit Hilfe von User Stories definiert.	User Story
ABC	include in the software requirements: usability engineering requirements that are sensitive to human errors and training	Anforderungen an die Gebrauchstauglichkeit werden über User Stories definiert.	User Story

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	include in the software requirements: user documentation to be developed	Zu erzeugende Anwender-Dokumentation wird als Anforderung im Product Backlog verwaltet.	Product Backlog
ABC	include in the software requirements: user maintenance requirements	Anforderungen an die Wartung durch den Anwender werden mit User Stories definiert.	User Story
BC	include RISK CONTROL measures implemented in software for hardware failures and potential software defects in the requirements	Im Sprint Planning Meeting wird sichergestellt, dass alle Maßnahmen aus der Risiko-Analyse einer in diesem Sprint geplanten Anforderung in das Product Backlog als Anforderung einfließen.	Risk Analysis and Management
ABC	re-EVALUATE the MEDICAL DEVICE RISK ANALYSIS when software requirements are established and update it as appropriate	Im Sprint Planning Meeting durchlaufen alle Anforderungen eine Risiko-Analyse, um notwendige Maßnahmen abzuleiten.	Risk Analysis and Management

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	Requirements are to be identified in the software requirements in such a way as to make it possible to trace the RISK CONTROL measures to the software requirements.	Eine User Story beinhaltet alle ermittelten Maßnahmen aus ihrer eigenen Risiko-Analyse.	User Story
ABC	software requirements: are expressed in terms that avoid ambiguity	Im Sprint Planning Meeting wird sichergestellt, dass über die Anforderungen ein einheitliches Verständnis bei allen Beteiligten herrscht.	Sprint Planning
ABC	software requirements: are stated in terms that permit establishment of test criteria and performance of tests to determine whether the test criteria have been met	User Stories sind so formuliert, dass sich Test- und Akzeptanz-Kriterien ableiten lassen.	User Story
ABC	software requirements: are traceable to SYSTEM requirements or other source	User Stories und darüberliegende bzw. externe Anforderungen werden mit Hilfe eindeutiger, referenzierbarer IDs versehen.	Linkable IDs

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	software requirements: can be uniquely identified	User Stories und darüberliegende bzw. externe Anforderungen werden mit Hilfe eindeutiger, referenzierbarer IDs versehen.	Linkable IDs
ABC	software requirements: do not contradict one another	Im Sprint Planning Meeting wird vom Team festgestellt, ob sich Anforderungen im Product Backlog widersprechen.	Sprint Planning
ABC	software requirements: implement SYSTEM requirements including those relating to RISK CONTROL	Das Product Backlog enthält alle Anforderungen, die sich aus dem Risikomanagement-Prozess ergeben.	Product Backlog
ABC	To demonstrate that the requirements have been implemented as desired, each requirement should be stated in such a way that objective criteria can be established to determine whether it has been implemented correctly.	Mit Hilfe von definierten Akzeptanz-Tests wird sichergestellt, dass das erstellte Software-Produkt die Bedürfnisse der Anwender befriedigt und die Anforderungen korrekt umgesetzt wurden.	Acceptance Testing

§ 5.3 - Software architectural design

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	define the major structural components of the software, their externally visible properties, and the relationship among them	Die grundlegenden Software-Strukturen und Architektur-Entscheidungen können im Sprint Planning getroffen werden.	Sprint Planning
BC	develop and document an ARCHITECTURE for the interfaces between the SOFTWARE ITEMS and the components external to the SOFTWARE ITEMS (both software and hardware), and between the SOFTWARE ITEMS	Im Sprint Planning werden die Architektur und die Schnittstellen der umzusetzenden Software-Einheiten sowie der einzubindenden externen Software-Einheiten festgelegt.	Sprint Planning
BC	Document the architecture so that it can be verified.	Mit der Architektur-Dokumentation ist es möglich, die Architektur zu verifizieren.	Architecture Documentation

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	during all ACTIVITIES of the software ARCHITECTURE, the classification of the SOFTWARE SYSTEM and SOFTWARE ITEMS should be re-EVALUATED	Die Sicherheitsklassifizierung wird im Sprint Planning für alle umzusetzenden Software-Einheiten überdacht und gegebenenfalls neu vergeben.	Risk Analysis and Management
C	identify the segregation between SOFTWARE ITEMS that is essential to RISK CONTROL, and state how to ensure that the segregation is effective	Im Sprint Planning werden die Architektur und die Software-Einheiten der umzusetzenden Anforderungen definiert. Die identifizierten Software-Einheiten durchlaufen die Risiko-Analyse.	Risk Analysis and Management
BC	specify functional and performance requirements for the SOUP item that are necessary for its intended use	Im Product Backlog Grooming werden die Funktionalitäten und Performance-Anforderungen an einzusetzende SOUP-Komponenten definiert.	Product Backlog Grooming
BC	specify functional and performance requirements for the SOUP item that are necessary for its intended use	Im Sprint Planning werden für benötigte SOUP-Komponenten die Funktionalitäten und Performance-Anforderungen festgelegt.	Sprint Planning

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	specify the SYSTEM hardware and software necessary to support the proper operation of the SOUP item	Im Backlog Grooming werden die Anforderungen an Hardware und Software identifiziert, welche durch den Einsatz von SOUP notwendig werden.	Product Backlog Grooming
BC	transform the requirements for the MEDICAL DEVICE SOFTWARE into a documented ARCHITECTURE that describes the software's structure	Durch die Anwendung von Test-Driven Development (TDD) entsteht während der Umsetzung von Anforderungen eine funktionierende Software-Architektur.	Test-Driven Development (TDD)
BC	transform the requirements for the MEDICAL DEVICE SOFTWARE into a documented ARCHITECTURE that describes the software's structure	Die umgesetzte Software-Architektur kann mit Hilfe von UML-Diagrammen beschrieben und dokumentiert werden.	Architecture Documentation

§ 5.4 - Software detailed design

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	define the level of detail of the SOFTWARE UNIT (can also be larger software items which are not decomposed)	Durch die Anwendung objekt-orientierter Prinzipien wird die Aufteilung von Software-Einheiten in weitere Einheiten getrieben.	Clean Code / SOLID Principles
ABC	define the level of detail of the SOFTWARE UNIT (can also be larger software items which are not decomposed)	Durch Test-Driven Development (TDD) wird direkt bei der Umsetzung von Software-Einheiten entschieden, wann und wie diese weiter untergliedert werden in einzelne Einheiten.	Test-Driven Development (TDD)
C	develop and document a detailed design for any interfaces between the SOFTWARE UNIT and external components (hardware or software), as well as any interfaces between SOFTWARE UNITS	Durch die Anwendung von Test-Driven Development (TDD) entsteht die Schnittstellspezifikation aller Software-Einheiten in Form von ausführbaren Tests gegen diese Schnittstellen.	Test-Driven Development (TDD)

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
C	develop and document a detailed design for each SOFTWARE UNIT of the SOFTWARE ITEM	Durch die Anwendung von Test-Driven Development (TDD) entsteht ein "Design by Contract" über die Schnittstellenspezifikation aller Software-Einheiten in Form von ausführbaren Tests gegen diese Schnittstellen.	Test-Driven Development (TDD)
BC	refine the SOFTWARE ITEMS and interfaces defined in the ARCHITECTURE to create SOFTWARE UNITS and their interfaces	Im Sprint Planning werden die umzusetzenden Anforderungen in eine Architektur überführt und diese wiederum in Software-Einheiten untergebrochen.	Sprint Planning
BC	refine the SOFTWARE ITEMS and interfaces defined in the ARCHITECTURE to create SOFTWARE UNITS and their interfaces	Durch den Einsatz von Test-Driven Development (TDD) entstehen Software-Einheiten und ihre Schnittstellen auf unterster Ebene, welche die darüberliegenden Software-Strukturen verfeinern.	Test-Driven Development (TDD)

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	SOFTWARE UNITS can be tested separately	Software-Einheiten dürfen mit Unit Tests getestet werden, ohne die Einbindung anderer Software-Einheiten.	Unit Testing
C	verify and document that the software detailed design implements and follows the software ARCHITECTURE	Im Sprint Review wird präsentiert, dass das umgesetzte Software-Design der Software-Architektur entspricht.	Sprint Review

§ 5.5 - Software unit implementation and verification

SK	Normtext	Herstellung der Normkonformität	Praktik/Artefakt
BC	ensure that SOFTWARE UNITS meet acceptance criteria	Durch Akzeptanztests wird sichergestellt, dass die entwickelte Software entsprechende Akzeptanzkriterien befriedigt.	Acceptance Testing
BC	ensure that SOFTWARE UNITS meet acceptance criteria	Im Sprint Review zeigt das Team, welche Akzeptanzkriterien in Form von Testfällen die Software erfolgreich durchlaufen hat.	Sprint Review
BC	ensure that SOFTWARE UNITS meet acceptance criteria	Durch Test-Automatation wird sichergestellt, dass die Software immer wieder die definierten Akzeptanzkriterien in Form von automatisierten Tests erfolgreich durchläuft.	Test Automation

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	ensure that SOFTWARE UNITS meet acceptance criteria	Durch Test-getriebene Entwicklung (TTD) wird sichergestellt, dass die entwickelte Software entsprechende Akzeptanzkriterien auf unterster Ebene befriedigt.	Test-Driven Development (TDD)
BC	establish acceptance criteria for SOFTWARE UNITS prior to integration into larger SOFTWARE ITEMS (risk control measures, interfaces, coding standards)	Mit Hilfe von Akzeptanz-Tests wird sichergestellt, dass eine in Software umgesetzte Anforderung abgekapselt funktioniert und damit für eine Integration in andere Strukturen bereit ist.	Acceptance Testing
BC	establish acceptance criteria for SOFTWARE UNITS prior to integration into larger SOFTWARE ITEMS (risk control measures, interfaces, coding standards)	Im Sprint Planning Meeting wird sichergestellt, dass zu jeder Anforderung Akzeptanz-Kriterien für die Integration in andere Strukturen erstellt werden.	Sprint Planning

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	establish strategies, methods and procedures for verifying each SOFTWARE UNIT. Where VERIFICATION is done by testing, the test procedures shall be EVALUATED for correctness.	Durch Test-getriebene Entwicklung wird die Software-Verifizierung sichergestellt. TDD ist ein definierter, korrekter Prozess.	Test-Driven Development (TDD)
C	include additional acceptance criteria as appropriate for: proper event sequence; data and control flow; planned resource allocation; fault handling; initialisation of variables; self-diagnostics; memory management; memory overflows; boundary conditio	Technische Akzeptanzkriterien, wie z.B. Sequenzen, Abläufe, Kontrollflüsse, Fehlerbehandlung, Initialisierung und Grenzwerte, werden mit Hilfe von Unit Tests definiert.	Unit Testing
BC	perform the SOFTWARE UNIT VERIFICATION and document the results	Durch Test-Automatation können die Ergebnisse aller Unit-Tests als Dokumentation erzeugt werden.	Test Automation

SK	Normtext	Herstellung der Normkonformität	Praktik/Artefakt
BC	perform the SOFTWARE UNIT VERIFICATION and document the results	Die Software wird durch Unit Tests auf unterster Ebene verifiziert. Durch Test-Automation kann eine Dokumentation der Testergebnisse erzeugt werden.	Unit Testing
ABC	write and verify the code for the SOFTWARE UNITS. The detailed design is to be translated into source code.	Mit Test-Driven Development (TDD) entsteht das Design des zu erstellenden Codes durch ausführbare Schnittstellen-Tests. Diese werden dann durch produktiven Code erfolgreich durchlaufen.	Test-Driven Development (TDD)

§ 5.6 - Software integration and integration testing

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	conduct REGRESSION TESTING appropriate to demonstrate that defects have not been introduced into previously integrated software	Regressionstests vorhandener Software-Versionen können durch Test-Automation jederzeit sicherstellen, dass keine Fehler im Software-System entstanden sind.	Test Automation
BC	document the test result (pass/fail and a list of ANOMALIES)	Mit einer Test-Automation kann durch die Durchführung aller vorhandenen Tests eine Dokumentation der Test-Resultate erzeugt werden.	Test Automation
BC	enter ANOMALIES found during software integration and integration testing into a software problem resolution PROCESS	Eine konsequente Zero-Bug-Tolerance führt dazu, dass alle Fehler, welche während der Software-Integration gefunden werden, schnellstmöglich behoben oder als irrelevant definiert werden.	Zero Bug Tolerance

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	enter ANOMALIES found during software integration and integration testing into a software problem resolution PROCESS	Fehler, welche während der Software-Integration auftauchen, werden im Sprint Backlog gepflegt, um sie schnellstmöglich zu beheben.	Sprint Development Plan = Sprint Backlog
BC	EVALUATE the integration test procedures for correctness	Im Sprint Review werden die Integrationstests auf Korrektheit geprüft.	Sprint Review
BC	identify the tester	Die Iteration Notes (Sprint Notizen) beinhalten die Liste der in der Iteration beteiligten Team-Mitglieder, wie zum Beispiel auch die Tester.	Iteration Notes (Sprint Notizen)
ABC	In order to fully test a SOFTWARE PRODUCT both black and white box testing might be required.	Akzeptanztests entsprechen Black-Box-Tests.	Acceptance Testing
ABC	In order to fully test a SOFTWARE PRODUCT both black and white box testing might be required.	Exploratives Testen entspricht Black-Box-Tests.	Exploratory and Manual Testing

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	In order to fully test a SOFTWARE PRODUCT both black and white box testing might be required.	Integrations-Tests entsprechen Black-Box-Tests bezüglich der Schnittstellen einzelner Systemkomponenten und White-Box-Tests bezüglich der System-Architektur.	Integration Testing
ABC	In order to fully test a SOFTWARE PRODUCT both black and white box testing might be required.	Unit Tests entsprechen Black-Box-Tests bezüglich der atomaren Schnittstellen und White-Box-Tests bezüglich der Software-Strukturen.	Unit Testing
BC	integrate the software units	Die einzelnen Software-Einheiten werden kontinuierlich integriert.	Continuous Integration
BC	retain sufficient records to permit the test to be repeated	Durch Test-Automation können bestehende Testfälle jederzeit wiederholt werden.	Test Automation

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	Software integration and system testing can be performed in a simulated environment, on actual target hardware, or on the full MEDICAL DEVICE.	Integrations- sowie System-Tests können mit Simulationen, Prototypen und Mocks durchgeführt werden.	Integration Testing
ABC	Software integration and system testing can be performed in a simulated environment, on actual target hardware, or on the full MEDICAL DEVICE.	Eine Anforderung oder User Story kann als erledigt betrachtet werden, wenn die entsprechenden Integrations- und System-Tests mit einer Simulation, einem Prototypen oder einem Mock durchgeführt wurden.	Definition of Done
BC	test the integrated SOFTWARE ITEMS in accordance with the integration plan and document the results	Die integrierten Software-Einheiten werden durch Integrationstests verifiziert. Durch Test-Automation kann eine Dokumentation der Testergebnisse erzeugt werden.	Integration Testing

SK	Normtext	Herstellung der Normkonformität	Praktik/Artefakt
BC	verify and record all aspects of the software integration in accordance with the integration plan	Die Software-Integration wird durch Integrationstests verifiziert.	Integration Testing

§ 5.7 - Software system testing

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	enter ANOMALIES found during software system testing into a software problem resolution PROCESS	Eine konsequente Zero-Bug-Tolerance führt dazu, dass alle Fehler, welche während der Software- und System-Tests gefunden werden, schnellstmöglich behoben oder als irrelevant definiert werden.	Zero Bug Tolerance
BC	enter ANOMALIES found during software system testing into a software problem resolution PROCESS	Fehler, welche während der System-Verifikation gefunden und nicht sofort behoben werden, werden in das Product Backlog einsortiert.	Product Backlog
BC	establish and perform a set of tests, expressed as input stimuli, expected outcomes, pass/fail criteria and procedures, for conducting SOFTWARE SYSTEM testing, such that all software requirements are covered	Durch Test-Automation können die Testfälle aller Software-Anforderungen jederzeit durchgeführt werden.	Test Automation

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	establish and perform a set of tests, expressed as input stimuli, expected outcomes, pass/fail criteria and procedures, for conducting SOFTWARE SYSTEM testing, such that all software requirements are covered	Eine User Story beinhaltet alle Testfälle, welche notwendig sind, um die Anforderungen dieser User Story prüfen und sicherstellen zu können.	User Story
ABC	If a decision has been made not to fix anomalies, they need to be EVALUATED in relation to the HAZARD analysis and the SAFETY of the device. Root cause and symptoms of ANOMALIES, and the rationale for not fixing them should be documented.	Bei einer Zero Bug Tolerance muss nicht jedes Problem und nicht jeder Fehler behoben werden. Falls entschieden wird, einen Fehler nicht zu beheben, muss die Begründung dokumentiert werden. Dies kann direkt im entsprechenden Backlog-Eintrag erfolgen.	Zero Bug Tolerance
BC	It is acceptable to combine integration testing and SOFTWARE SYSTEM testing into a single plan.	Im Sprint Planning erfolgt die Planung der Integrations-Tests sowie der System-Tests.	Sprint Planning

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	It is acceptable to test software requirements in earlier phases.	Anforderungen können bereits vor ihrer endgültigen Umsetzung getestet und integriert werden.	Early Integration / Early Testing
ABC	test coverage of requirements, RISK CONTROL, usability, and test types (e.g., fault, installation, stress) should be demonstrated and documented	Im Sprint Review werden die entwickelten und durchlaufenen Tests, sowie die Test-Abdeckung demonstriert.	Sprint Review
BC	verify that all software requirements have been tested or otherwise VERIFIED	Im Sprint Review Meeting berichtet das Team über die durchgeführten Tests aller umgesetzten Anforderungen.	Sprint Review
BC	verify that all software requirements have been tested or otherwise VERIFIED	Mit Hilfe einer umfassenden Test-Automation kann sichergestellt und nachgewiesen werden, dass alle Anforderungen getestet und verifiziert sind.	Test Automation

SK	Normtext	Herstellung der Normkonformität	Praktik/Artefakt
BC	verify that SOFTWARE SYSTEM test procedures trace to software requirements	Im Sprint Review Meeting präsentiert das Team die erzeugten und durchgeführten Tests zu den umgesetzten Anforderungen. (Es werden keine Tests erstellt, zu denen keine Anforderungen existieren.)	Sprint Review
BC	verify that SOFTWARE SYSTEM test procedures trace to software requirements	User Stories beinhalten neben der Definition einer Anforderung auch deren Akzeptanz- und Testkriterien.	User Story
BC	verify that test results meet the required pass/fail criteria	Im Sprint Review werden die umgesetzten Anforderung auf Basis ihrer Akzeptanzkriterien abgenommen und freigegeben.	Sprint Review
BC	verify that the VERIFICATION strategies and the test procedures used are appropriate	Im Sprint Review wird über Verifikation und Testen berichtet.	Sprint Review

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	verify that the VERIFICATION strategies and the test procedures used are appropriate	Die Definition-of-Done stellt sicher, dass Verifikation und entsprechende Test-Prozeduren eingehalten werden.	Definition of Done
ABC	When a change is made to a SOFTWARE SYSTEM, REGRESSION TESTING should be determined, planned and documented.	Durch Test-Automatation kann das gesamte Software-System nach einer Änderung vollständig mit allen vorhandenen Testfällen geprüft werden.	Test Automation
BC	When changes are made during SOFTWARE SYSTEM testing, conduct testing appropriate to demonstrate that unintended side effects have not been introduced	Durch Test-Automatation kann das gesamte Software-System nach einer Änderung vollständig mit allen vorhandenen Testfällen geprüft werden.	Test Automation
BC	When changes are made during SOFTWARE SYSTEM testing, perform relevant RISK MANAGEMENT ACTIVITIES	Notwendige Änderungen an der Software, die durch System-Tests erkannt werden, durchlaufen während des Sprint Plannings die notwendige Risikoanalyse.	Risk Analysis and Management

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	When changes are made during SOFTWARE SYSTEM testing, repeat tests, perform modified tests or perform additional tests, as appropriate, to verify the effectiveness of the change in correcting the problem	Durch Test-Automation können alle vorhandenen Testfälle des Software-Systems nach Änderungen jederzeit durchgeführt werden.	Test Automation

§ 5.8 - Software release

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	archive the SOFTWARE PRODUCT, the CONFIGURATION ITEMS, and the documentation	Durch ein Versionierungssystem werden alle Bestandteile eines Software-Release archiviert.	Versioning System
BC	document all known residual ANOMALIES	Nicht behobene Fehler und Abweichungen werden in den Iterations-Notizen festgehalten.	Iteration Notes (Sprint Notizen)
BC	document the procedure and environment used to create the released software	Der Software-Entwicklungsplan beschreibt die Vorgehensweise zur Erstellung eines Software-Release, sowie das dafür notwendige Environment.	Software Development Plan
ABC	document the VERSION of the SOFTWARE PRODUCT that is being released	In den Release Notes ist die Version des Software-Release enthalten.	Release Notes

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	ensure that all ACTIVITIES and TASKS are complete along with all the associated documentation	Durch die Definition-of-Done wird sichergestellt, dass alle Aufgaben zum Zeitpunkt eines Software-Release abgeschlossen sind.	Definition of Done
BC	ensure that software VERIFICATION has been completed and the results EVALUATED before the software is released	Im Sprint Review werden die umgesetzten Anforderungen evaluiert und freigegeben, bevor sie in ein Software-Release gelangen können.	Sprint Review
BC	ensure that software VERIFICATION has been completed and the results EVALUATED before the software is released	In der Definition-of-Done wird festgelegt, dass die Software bzw. jede User Story verifiziert sein muss, bevor sie ausgeliefert werden darf.	Definition of Done
BC	ensure that the released SOFTWARE PRODUCT can be reliably delivered: replication, media labelling, packaging, protection, storage, and delivery.	Im Release Planning wird entschieden, welche Maßnahmen und Aktivitäten durchzuführen sind, um eine verlässliche Auslieferung der Software zu gewährleisten.	Release Planning Meeting

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	ensure that the released SOFTWARE PRODUCT can be reliably delivered: replication, media labelling, packaging, protection, storage, and delivery.	Im Product Backlog sind Maßnahmen und Aktivitäten enthalten, die eine verlässliche Auslieferung der Software gewährleisten.	Product Backlog
BC	EVALUATE known residual ANOMALIES to ensure that they do not contribute to an unacceptable RISK	Bei Anwendung der Zero Bug Tolerance durchläuft jeder Fehler und jede Abweichung die Risiko-Analyse. Nur wenn ein Fehler und eine Abweichung nicht zu einem inakzeptablen Risiko beitragen, darf entschieden werden, den Fehler und die Abweichung nicht zu beheben.	Zero Bug Tolerance

§ 6 - Software maintenance process

§ 6.1 - Establish software maintenance plan

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	establish a software maintenance plan for conducting the ACTIVITIES and TASKS of the maintenance PROCESS	Im Software Maintenance Plan werden alle Aktivitäten und Aufgaben des Maintenance Prozesses beschrieben.	Software Maintenance Plan
ABC	software maintenance plan includes criteria for determining whether feedback is considered to be a problem	Im Software Maintenance Plan wird festgelegt, anhand welcher Kriterien ein Feedback als Problem/Fehler betrachtet wird.	Software Maintenance Plan
ABC	software maintenance plan includes procedures for: receiving, documenting, evaluating, resolving and tracking feedback arising after release of the MEDICAL DEVICE SOFTWARE	Im Software Maintenance Plan wird festgelegt, wie nach einem Release das Feedback eingeholt, dokumentiert, ausgewertet, umgesetzt und verfolgt wird.	Software Maintenance Plan

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	software maintenance plan includes use of the software configuration management PROCESS for managing modifications to the existing SYSTEM	Im Software Maintenance Plan wird der Einsatz des Versionierungs-Systems beschrieben.	Software Maintenance Plan
ABC	software maintenance plan includes use of the software problem resolution PROCESS for analysing and resolving problems arising after release of the MEDICAL DEVICE SOFTWARE	Im Product Backlog Grooming werden Probleme und Fehler von bereits releaster Software besprochen.	Product Backlog Grooming
ABC	software maintenance plan includes use of the software problem resolution PROCESS for analysing and resolving problems arising after release of the MEDICAL DEVICE SOFTWARE	Im Release Planning findet die Vorbereitung und Priorisierung statt für Probleme und Fehler von bereits releaster Software.	Release Planning Meeting

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	software maintenance plan includes use of the software problem resolution PROCESS for analysing and resolving problems arising after release of the MEDICAL DEVICE SOFTWARE	Im Sprint Planning finden die notwendigen Risiko-Analysen sowie die Umsetzungsplanung statt für Probleme und Fehler von bereits releaster Software.	Sprint Planning

§ 6.2 - Problem and modification analysis

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	analyse each CHANGE REQUEST for its effect on the organization, released SOFTWARE PRODUCTS, and SYSTEMS with which it interfaces	Im Product Backlog Grooming Meeting werden neue und geänderte Anforderungen auf ihren Einfluß auf bestehende Produkte und Systeme untersucht.	Product Backlog Grooming
ABC	EVALUATE and approve CHANGE REQUESTS which modify released SOFTWARE PRODUCTS	Der Product Owner läßt im Product Backlog Grooming Meeting neue und geänderte Anforderungen vom Team bewerten und erteilt deren Freigabe durch Aufnahme und Einsortierung in sein Product Backlog.	Product Backlog Grooming

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	EVALUATE Each PROBLEM REPORT to determine how it affects the SAFETY of a released SOFTWARE PRODUCT and whether a change to the released SOFTWARE PRODUCT is needed to address the problem	Im Product Backlog Grooming Meeting werden Fehler- und Problem-Berichte bewertet und ihr Einfluß auf die Sicherheit bestehender Produkte und Systeme festgestellt, um notwendige Anforderungsänderungen im Product Backlog abzuleiten.	Product Backlog Grooming
ABC	inform users and regulators about any problem in released SOFTWARE PRODUCTS and the consequences of continued unchanged use	Probleme und deren Konsequenzen werden über einen "Problem Communication"-Prozess an Anwender kommuniziert.	Problem Communication
ABC	inform users and regulators about: the nature of any available changes to released SOFTWARE PRODUCTS and how to obtain and install the changes	Die Release Notes einer Software beinhalten alle Änderungen an dem Software-System seit dem letzten Release.	Release Notes

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	monitor feedback on released SOFTWARE PRODUCT from inside its own organization	Über den Software Maintenance Prozess erfolgt das Einholen des Feedbacks zu releaster Software innerhalb der eigenen Organisation.	Software Maintenance Process
ABC	monitor feedback on released SOFTWARE PRODUCT from users	Über den Software Maintenance Prozess erfolgt das Einholen des Feedbacks zu releaster Software von tatsächlichen Anwendern.	Software Maintenance Process
ABC	record PROBLEM REPORTS that include actual or potential adverse events, and deviations from specifications	User Stories können auch Fehler- und Problembenrichte darstellen, welche Abweichungen der Anforderungen und der Spezifikation beinhalten.	User Story

§ 6.3 - Modification implementation

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	Modifications may be released as part of a full re-release of a SOFTWARE SYSTEM or as a modification kit comprising changed SOFTWARE ITEMS and the necessary tools to install the changes as modifications to an existing SOFTWARE SYSTEM	Im Release Planning wird festgelegt, ob ein Release als vollständiges System-Release erstellt wird oder als Patch zu einem bestehenden System-Release.	Release Planning Meeting
ABC	release modified SOFTWARE SYSTEMS according to 5.8 (Software Release)	Das Release Planning unterscheidet nicht, ob ein Release aus neuen Anforderungen besteht oder aus Änderungen an einem bestehenden System.	Release Planning Meeting

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	use the software development PROCESS or an established maintenance PROCESS to implement the modifications	Im Product Backlog Grooming werden Änderungen an der Software analysiert, bewertet und ins Product Backlog einsortiert. Damit durchlaufen Änderungen den gleichen Entwicklungsprozess wie Anforderungen und User Stories.	Product Backlog Grooming

§ 7 - Software risk management process

§ 7.1 - Analysis of software contributing to hazardous situations

SK	Normtext	Herstellung der Normkonformität	Praktik/Artefakt
ABC	all device hazardous situations can be identified in parallel to definition of the software ARCHITECTURE	Im Sprint Planning wird die Risiko-Analyse der umzusetzenden Architektur und ihrer Software-Einheiten durchgeführt.	Risk Analysis and Management
BC	consider potential causes including failure or unexpected results from SOUP	Im Sprint Planning werden mögliche Ursachen für Fehlverhalten von SOUP durch die Risiko-Analyse identifiziert.	Risk Analysis and Management
BC	consider potential causes including hardware failures or other software defects that could result in unpredictable software operation	Im Sprint Planning werden für fehlerhaftes Verhalten der Software mögliche Ursachen in Hardware- und Software-Fehlern analysiert.	Risk Analysis and Management

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	consider potential causes including incorrect or incomplete specification of functionality	Spätestens im Sprint Planning Meeting werden Anforderungen auf ihre korrekte und vollständige Definition geprüft.	Sprint Planning
BC	consider potential causes including incorrect or incomplete specification of functionality	Die Definition-of-Ready einer Anforderung fordert deren korrekte und vollständige Definition.	Definition of Ready
BC	consider potential causes including reasonably foreseeable misuse	In der Risiko-Analyse während des Sprint Planning Meetings werden alle Anforderungen auf möglichen Mißbrauch untersucht und entsprechende Maßnahmen abgeleitet.	Risk Analysis and Management
BC	consider potential causes including software defects in the identified SOFTWARE ITEM functionality	Im Sprint Planning werden die umzusetzenden Anforderungen und deren abgeleitete Software-Einheiten durch die Risiko-Analyse auf mögliche Gefährdungen hin untersucht.	Risk Analysis and Management

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	document in the RISK MANAGEMENT FILE potential causes of the SOFTWARE ITEM contributing to a hazardous situation	Im Sprint Planning werden die umzusetzenden Software-Einheiten durch die Risiko-Analyse auf Gefahrensituationen untersucht.	Risk Analysis and Management
BC	document in the RISK MANAGEMENT FILE sequences of events that could result in a hazardous situation	Im Sprint Planning werden Workflows und sequentielle Abfolgen von Events in der Software durch die Risiko-Analyse auf Gefahrensituationen untersucht.	Risk Analysis and Management
BC	evaluate SOUPs and their list of known anomalies that could lead to a hazardous situation	Im Sprint Planning werden SOUPs und ihre bekannten Fehler/Abweichungen durch die Risiko-Analyse betrachtet.	Risk Analysis and Management
BC	identify SOFTWARE ITEMS that could contribute to a hazardous situation identified in the MEDICAL DEVICE RISK ANALYSIS ACTIVITY of ISO 14971	Im Sprint Planning werden User Stories daraufhin untersucht, ob sie zu einer Gefahrensituation beitragen können, die in der Risiko-Analyse enthalten ist.	Risk Analysis and Management

§ 7.2 - Risk control measures

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	assign a software safety class to the SOFTWARE ITEM based on the possible effects of the HAZARD that the RISK CONTROL measure is controlling	Im Sprint Planning wird der zu erstellenden Software-Einheit eine Sicherheitsklassifizierung zugewiesen (A, B, C), welche weitere Details im Entwicklungsprozess bestimmt.	Risk Analysis and Management
BC	define and document risk control measures for each potentially hazardous software item	Im Sprint Planning werden zu jeder Software-Einheit mit möglichen Gefahrensituationen entsprechende Maßnahmen identifiziert und festgehalten.	Risk Analysis and Management
BC	develop the SOFTWARE ITEM implementing a RISK CONTROL measure in accordance with Clause 5 (Software Development Process)	Im Sprint Planning werden Software-Einheiten, die eine Risiko-Maßnahme umsetzen, genauso behandelt wie alle anderen Anforderungen und befolgen den definierten Entwicklungsprozess.	Sprint Planning

SK	Normtext	Herstellung der Normkonformität	Praktik/Artefakt
BC	include the RISK CONTROL measure in the software requirements	Im Sprint Planning Meeting wird sichergestellt, dass alle Maßnahmen aus der Risiko-Analyse einer in diesem Sprint geplanten Anforderung in das Product Backlog als Anforderung einfließen.	Risk Analysis and Management

§ 7.3 - Verification of risk control measures

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	document the verification of implemented risk control measures	Die Iteration Notes (Sprint Notizen) beinhalten die umgesetzten Maßnahmen aus der Risikoanalyse und deren Verifizierung im aktuellen Sprint.	Iteration Notes (Sprint Notizen)
BC	document TRACEABILITY of software HAZARDS	Mit Hilfe von verweisenden IDs kann die Traceability hergestellt werden von erkannten Gefahrensituationen über Maßnahmen über umzusetzende Software-Einheiten bis zur Verifikation dieser Software-Einheiten.	Linkable IDs
BC	evaluate implemented risk control measures to identify new possible hazards	Im Sprint Review werden umgesetzte Risiko-Maßnahmen auf neue mögliche Gefahrensituationen hin untersucht.	Sprint Review

SK	Normtext	Herstellung der Normkonformität	Praktik/Artefakt
BC	verify the implementation of each RISK CONTROL measure	Im Sprint Review wird die Umsetzung aller Risiko-Maßnahmen berichtet und verifiziert.	Sprint Review

§ 7.4 - Risk management of software changes

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
BC	analyse changes to the MEDICAL DEVICE SOFTWARE (including SOUP) to determine interference with existing risk control measures	Im Product Backlog Grooming Meeting durchlaufen neue und geänderte Anforderungen den Risikomanagement-Prozess, um sie mit bestehenden Risiko-Maßnahmen abzugleichen.	Product Backlog Grooming
ABC	analyse changes to the MEDICAL DEVICE SOFTWARE (including SOUP) to determine new potential hazards and risk control measures	Im Product Backlog Grooming Meeting durchlaufen neue und geänderte Anforderungen den Risikomanagement-Prozess, um neue Risiko-Maßnahmen abzuleiten.	Product Backlog Grooming

§ 8 - Software configuration management process

§ 8.1 - Configuration identification

SK	Normtext	Herstellung der Normkonformität	Praktik/Artefakt
ABC	document the set of CONFIGURATION ITEMS and their VERSIONS that comprise the SOFTWARE SYSTEM configuration	Alle zu einem Software-System gehörenden Bestandteile und deren Versionen werden in einem Versionierungs-System abgelegt. Aus diesem Versionierungs-System kann die gültige, vollständige System-Konfiguration aller Bestandteile festgestellt werden.	Versioning System
ABC	document the title, the MANUFACTURER, and the unique SOUP designator of each SOUP CONFIGURATION ITEM being used	Die im Software-System eingesetzten Dritt-Party-Komponenten werden in ihrem Ursprungszustand mit einem Versionierungs-System verwaltet.	Versioning System

SK	Normtext	Herstellung der Normkonformität	Praktik/Artefakt
ABC	establish a scheme for the unique identification of CONFIGURATION ITEMS and their VERSIONS	Mit einem Versionierungssystem können alle zu einem Software-System gehörenden Komponenten und deren Versionen identifiziert und verwaltet werden.	Versioning System

§ 8.2 - Change control

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	change CONFIGURATION ITEMS only in response to an approved CHANGE REQUEST	Die Umsetzung von Anforderungen, welche das bestehende Software-System verändern (in einem inkrementell-iterativen Prozess also alle Anforderungen), muss vom Product Owner beauftragt sein. Dies kann mit Hilfe der Definition-of-Ready sichergestellt werden.	Definition of Ready
ABC	identify and perform any ACTIVITY that needs to be repeated as a result of the change, including changes to the software safety classification of SOFTWARE SYSTEMS and SOFTWARE ITEMS	Im Sprint Planning Meeting werden vom Team alle Tasks identifiziert, welche für die Umsetzung einer Änderung (erneut) durchgeführt werden müssen.	Sprint Planning

SK	Normtext	Herstellung der Normkonformität	Praktik/Artefakt
ABC	provide traceability of change requests, their approvals, and relevant problem reports	Eine User Story, welche auf Basis eines Problems formuliert wird, beinhaltet den zugehörigen Problembericht, um Traceability herzustellen.	User Story
ABC	verify the change, including repeating any VERIFICATION that has been invalidated by the change	Durch Test-Automatation kann das Software-System jederzeit nach einer Änderung verifiziert werden.	Test Automation

§ 8.3 - Configuration status accounting

SK	Normtext	Herstellung der Normkonformität	Praktik/Artefakt
ABC	retain retrievable records of the history of controlled CONFIGURATION ITEMS including SYSTEM configuration	Mit Hilfe des Versionierungs-Systems kann die Historie aller Bestandteile des Software-Systems nachvollzogen werden.	Versioning System

§ 9 - Software problem resolution process

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	advise relevant parties of the existence of the problem	Einzubindende Stakeholder werden durch einen "Problem Communication"-Prozess über die Existenz eines Problems informiert.	Problem Communication
ABC	approve and implement all CHANGE REQUESTS, observing the requirements of the change control PROCESS	Im Product Backlog Grooming werden notwendige Änderungen am System diskutiert und für die Entwicklung freigegeben.	Product Backlog Grooming
ABC	create a CHANGE REQUEST(S) for actions needed to correct the problem, or document the rationale for taking no action	User Stories können Fehlermeldungen, Change Requests, sowie andere Einträge enthalten, um das Systemverhalten zu ändern oder zu korrigieren.	User Story

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	document the outcome of the investigation and evaluation	Die Ergebnisse von Fehleranalysen und -bewertungen werden in der User Story bzw. dem Fehlereintrag direkt festgehalten.	User Story
ABC	EVALUATE the problem's relevance to SAFETY using the software RISK MANAGEMENT PROCESS	Im Sprint Planning Meeting werden Probleme und Fehler auf sicherheitsrelevante Aspekte mit Hilfe der Risikoanalyse untersucht.	Risk Analysis and Management
ABC	for each problem detected prepare a PROBLEM REPORT classified as follows: type, scope, criticality.	In User Stories können Fehler- und Problembereiche beschrieben sein.	User Story
ABC	include in the test documentation: test results; ANOMALIES found; VERSION of software tested; relevant hardware and software test configurations; relevant test tools; date tested; identification of the tester	In der Test-Dokumentation werden folgende Informationen abgelegt: Test-Ergebnisse, gefundene Abweichungen, getestete Software-Version, Hardware- und Software-Konfigurationen, eingesetzte Test-Tools, Test-Datum, Test-Personal	Test Documentation

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	investigate the problem and if possible identify the causes	Im Sprint Planning werden die Fehler- und Problemberrichte analysiert.	Sprint Planning
ABC	maintain records of PROBLEM REPORTS and their resolution including their VERIFICATION	In den Iteration Notes (Sprint Notizen) sind alle Fehlerberichte aus externen Quellen enthalten, sowie deren abgeleitete Tests zur Verifikation.	Iteration Notes (Sprint Notizen)
ABC	maintain records of PROBLEM REPORTS and their resolution including their VERIFICATION	Fehlerberichte werden im Sprint Backlog verwaltet, um schnellstmöglich behoben zu werden.	Sprint Development Plan = Sprint Backlog
ABC	perform analysis to detect trends in PROBLEM REPORTS	Im Sprint Review werden die Problem- und Fehlermeldungen auf Trends und Tendenzen untersucht.	Sprint Review
ABC	verify resolutions to determine whether additional problems have been introduced	Im Sprint Review wird berichtet, welche neuen Probleme und Fehler gefunden wurden.	Sprint Review

SK	Normtext	Herstellung der Normkonformität	Praktik/ Artefakt
ABC	verify resolutions to determine whether adverse trends have been reversed	Im Sprint Review wird berichtet, ob Problemlösungen dazu geführt haben, nachteilige Entwicklungen umzukehren.	Sprint Review
ABC	verify resolutions to determine whether CHANGE REQUESTS have been implemented in the appropriate SOFTWARE PRODUCTS and ACTIVITIES	Im Sprint Review wird berichtet, welche Änderungen (Change Requests) an der Software durchgeführt wurden.	Sprint Review
ABC	verify resolutions to determine whether problem has been resolved and the PROBLEM REPORT has been closed	Im Sprint Review wird berichtet, welche Fehler und Probleme behoben wurden.	Sprint Review

Literatur- und Quellenverzeichnis

Normen

- IEC 62304 - Edition 1.0 - 2006-05
Medical device software - Software life cycle processes
- IEC 62366 - Edition 1.0 - 2007-10
Medical devices - Application of usability engineering to medical devices
- ISO 13485:2003
Medical devices - Quality management systems - Requirements for regulatory purposes
- ISO 14971:2007
Medical devices - Application of risk management to medical devices

Bücher

- Beck, Kent - Test-Driven Development
Addison-Wesley, 2002
- Cohn, Mike - User Stories Applied
Addison-Wesley, 2004
- Crispin, Lisa; Gregory, Janet - Agile Testing
Addison Wesley, 2008
- Derby, Esther; Larsen, Diana - Agile Retrospectives
Pragmatic Programmers, 2006

- Kerth, Norman - Project Retrospectives
Computer Bookshops, 2001
- Kua, Patrick - The Retrospectives Handbook
(<https://leanpub.com/the-retrospective-handbook>)
Leanpub, 2012
- Martin, Robert C. - Clean Code
Prentice Hall, 2008
- Osherove, Roy - The Art of Unit Testing
mitp, 2010
- Pichler, Roman - Scrum
dpunkt.verlag, 2007
- Schelle, Ottmann, Pfeiffer - ProjektManager (beinhaltet das gesamte Projektmanagement-Kompendium der GPM/IPMA)
GPM Deutsche Gesellschaft für Projektmanagement, 2008
- Schwaber, Ken - Agiles Projektmanagement mit Scrum
Microsoft Press Deutschland, 2007
- Shore, James; Warden, Shane - The Art of Agile Development
(Kapitel 13.2)
O'Reilly Media, 2007

Weblinks

- agile.dzone.com/news/norms-values-working
- artofunittesting.com/
- butunclebob.com/ArticleS.UncleBob.TheThreeRulesOfTdd
- c2.com/cgi/wiki?IntegrationTest
- c2.com/cgi/wiki?TestDrivenDevelopment

- c2.com/cgi/wiki?UnitTest
- de.slideshare.net/codecentric/exploratory-testing-inagileoverviewmeettheexpertselisabethhendrickson
- de.wikipedia.org/wiki/Clean_Code
- de.wikipedia.org/wiki/ISO_13485
- efasoft.blogspot.de/2010/10/software-architecture-documentation-in.html
- en.wikipedia.org/wiki/Acceptance_testing
- en.wikipedia.org/wiki/Continuous_integration
- en.wikipedia.org/wiki/Exploratory_testing
- en.wikipedia.org/wiki/Integration_testing
- en.wikipedia.org/wiki/Unit_testing
- guide.agilealliance.org/guide/acceptance.html
- guide.agilealliance.org/guide/backlog.html
- guide.agilealliance.org/guide/ci.html
- guide.agilealliance.org/guide/exploratory.html
- guide.agilealliance.org/guide/grooming.html
- guide.agilealliance.org/guide/heartbeatretro.html
- guide.agilealliance.org/guide/integration.html
- guide.agilealliance.org/guide/iteration.html
- guide.agilealliance.org/guide/ready.html
- guide.agilealliance.org/guide/sashimi.html
- guide.agilealliance.org/guide/stories.html
- guide.agilealliance.org/guide/tdd.html
- guide.agilealliance.org/guide/unittest.html
- guide.agilealliance.org/guide/usability.html
- sdt.bz/32548
- www.boost.co.nz/blog/agile/acceptance-criteria/
- www.clean-code-developer.de/SOLID.ashx

- www.cleancoders.com/
- www.energizedwork.com/weblog/2006/04/release-planning.html
- www.extremeprogramming.org/rules/planninggame.html
- www.extremeprogramming.org/rules/unittests.html
- www.extremeprogramming.org/rules/userstories.html
- www.ibm.com/developerworks/mydeveloperworks/blogs/c914709e-8097-4537-92ef-8982fc416138/entry/march_10_2012_11_27_am8?lang=en
- www.ibm.com/developerworks/rational/library/early-integration-testing-enables-agile-development/
- www.infoq.com/articles/agile-development-team-charter
- www.infoq.com/news/2011/02/zero-defect-systems
- www.joelonsoftware.com/articles/fog0000000043.html
(Punkt 5)
- www.leadingagile.com/2012/08/simple-cheat-sheet-to-sprint-planning-meeting/
- www.martinfowler.com/articles/continuousIntegration.html
- www.mindtools.com/pages/article/newTMM_95.htm
- www.mountangoatsoftware.com/scrum/product-backlog
- www.mountangoatsoftware.com/scrum/sprint-backlog
- www.mountangoatsoftware.com/scrum/sprint-planning-meeting
- www.mountangoatsoftware.com/scrum/sprint-review-meeting
- www.mountangoatsoftware.com/topics/user-stories
- www.oose.de/teamblog/team/frag-das-team-fur-architektur-dokumentation/

- www.romanpichler.com/blog/product-backlog/grooming-the-product-backlog/
- www.romanpichler.com/blog/product-backlog/the-definition-of-ready/
- www.satisfice.com/articles/et-article.pdf
- www.scrum.org/Resources/Scrum-Glossary/Definition-of-Done
- www.scrum.org/Resources/What-is-Scrum/
- www.scrum.org/Scrum-Guides
- www.scrumalliance.org/articles/105-what-is-definition-of-done-dod
- www.scrumalliance.org/articles/122--tips-for-creating-a-good-sprint-backlog
- www.scrumalliance.org/articles/339-how-to-hold-an-effective-backlog-grooming-session
- www.scrumcrazy.com/Tips+for+a+Good+Sprint+Review
- xprogramming.com/articles/automatedtesting/
- xprogramming.com/blog/automating-story-tests/

Feedback und Kontakt

Nach der Lektüre dieses Buches und der Anwendung seiner Inhalte auf Ihr konkretes Umfeld in Teams und Unternehmen bin ich sehr an Ihren Erfahrungen interessiert.

- Wie konnten Sie Scrum in Ihrem Umfeld umsetzen?
- An welchen Stellen hatten oder haben Sie Schwierigkeiten mit der Umsetzung?
- Was hat nicht funktioniert?
- Was hat gut funktioniert?
- Wo haben Sie Veränderungen vorgenommen?
- Mit welchen Change Management Methoden haben Sie diese Veränderungen vorgenommen?
- Welche Fragen sind unbeantwortet geblieben?

Feedback und konstruktive Kritik zu diesem Buch sind jederzeit willkommen. Sie können mich über meine Webseite www.agilecoach.de kontaktieren. Ich werde versuchen, Ihre Kommentare aufzunehmen und Ihre Fragen zu beantworten.

Vielen Dank für das Lesen dieses Buches.