

Teil VII

Anhang

A Die 12 Prinzipien hinter dem Agilen Manifest

An dieser Stelle werden die 12 Prinzipien des Agilen Manifest (*Agile Manifesto*), welche von den Gründern der agilen Bewegung erstellt wurden, aufgelistet. Zu finden sind diese Prinzipien, wie auch das Manifest selbst auf in vielen Sprachen auf der Webseite <http://www.agilemanifesto.org/>. Die folgenden Prinzipien stammen aus der deutschen Übersetzung (siehe [25]).

1. Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.
2. Heisse Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
3. Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
4. Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.
5. Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.
6. Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteam zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.
7. Funktionierende Software ist das wichtigste Fortschrittsmaß.
8. Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.
9. Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
10. Einfachheit – die Kunst, die Menge nicht getaner Arbeit zu maximieren – ist essenziell.
11. Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
12. In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.

B Ziele und Fragen der SSELab-Anforderungsanalyse

Informationsbedürfnisse, die anhand der Anforderungsanalyse durch SSELab-Projektleiter erhoben wurden, werden folglich anhand von Fragestellungen dieser dargestellt.

Fragen, welche mit [Z] markiert wurden, können zum aktuellen Zeitpunkt ohne weiteren Anpassungsaufwand der Mess-Tools bzw. der Entwicklungsprozesse nicht direkt integriert werden, wurden jedoch auch als wichtig bzw. interessant identifiziert.

B.1 Projektqualität

B.1.1 Planungssicherheit

- Verläuft das Projekt nach Plan?
- Können Projekttermine eingehalten werden?
- Wie sieht der aktuelle und voraussichtliche Arbeitsaufwand bis zum nächsten Meilenstein aus? [Z]
- Werden alle geplanten Features bis zum Meilenstein fertig gestellt? [Z]
- Werden alle offenen Tickets bis zum Meilenstein fertig bearbeitet? [Z]
- Wie viel Entwicklungsaufwand ist für eine bestimmte Aufgabe prognostiziert? [Z]
- Welche Mitarbeiter sind für das Projekt verfügbar? [Z]
- Wie verteilt sich die Verfügbarkeit der Mitarbeiter auf die vorhandenen Aufgaben auf? [Z]
- Wie viel Arbeitszeit steht für jeden Mitarbeiter noch zur Verfügung? [Z]

B.1.2 Projektleistung

- Wie verläuft die Ticketabarbeitung innerhalb des Projekts?
- Wie lange dauert die durchschnittliche Ticketabarbeitung?
- Wie viel Aufwand hat die Abarbeitung der Aufgaben/ Tickets tatsächlich gekostet? [Z]

- Wie verläuft die Abarbeitung der To Dos? [Z]
- Wie verläuft die Abarbeitung offener Fehler?
- Wie entwickeln sich die Fehler innerhalb der Projektlaufzeit?
- Wie hat sich die Anzahl der Tests innerhalb des Projekts entwickelt?
- Wie verläuft die Entwicklung des Testerfolgs?
- Wie entwickelt sich die Testabdeckung?
- Wie verlaufen die Builds im Allgemeinen?

B.2 Produktqualität

B.2.1 Prüfbarkeit

- Wie hoch ist der aktuelle Testerfolg?
- Wie hat sich der Testerfolg nach dem letzten Commit verändert?
- Wie hoch ist die aktuelle Testabdeckung?
- Wie hoch ist die Testabdeckung gegenüber den Anforderungen? [Z]
- Welche Commits haben den Testerfolg bzw. die Testabdeckung verschlechtert?

B.2.2 Reife

- Wie hoch ist die aktuelle Zahl der offenen Fehler?
- Wie viele kritischen Fehler existieren im Projekt?
- Wie verteilen sich die Fehler nach Härtegrad?
- Wie sieht die Verteilung der Fehler nach Priorität aus?
- Wie verteilen sich die Fehler bezüglich des Abarbeitungsstatus?

B.2.3 Änderbarkeit

- Wie hoch ist die Dichte der Kommentare innerhalb des Quellcodes?
- Besitzt jede Methode/ Klasse einen Kommentar?
- Wie hoch ist der Verletzungsgrad der Kodierrichtlinien?
- Welches sind die Top-5-Regelverletzungen innerhalb dieser Richtlinien?
- Wie viele Commits enthalten keine Ticket ID? [Z]

- Wurden Entwurfsmuster korrekt angewendet? [Z]
- Wie sieht die Abhängigkeit innerhalb von Klassen/ Paketen/ Komponenten aus? [Z]

B.2.4 Zuverlässigkeit

- Wie verläuft der Build?
- Welche Probleme treten beim Build auf?
- Welcher Commit hat Probleme in Build verursacht?

B.3 Zusätzliche Fragen [Z]

Folgende Fragestellungen wurden innerhalb der Interviews für interessant gehalten, können jedoch aufgrund der technischen oder projektinternen Voraussetzungen in diesem Stadium der Dashboard-Integration nicht realisiert werden.

- Was sind die neusten Änderungen im Projekt? (z.B. neue Tickets, Commits, Ankündigungen)
- Bei wie vielen Commits steht ein Review aus?
- Wie verläuft die Abarbeitung der Reviews?
- Wie viele Review-Items wurden abgelehnt?
- Existieren Codeplagiate und wenn ja, wie viele?
- Existieren Verletzungen der vorgegebenen Architektur?
- Wie sieht die aktuelle Anzahl der vorhandenen Branches aus?
- Wie verlaufen die Builds und Tests auf den jeweiligen Branches?
- Wie hoch ist der aktuelle Aufwand einer Branch-Integration in die Master-Umgebung?
- Wie sehen die Abhängigkeiten innerhalb der Projekte aus?
- Liegt der Build-Fehlschlag an einem internen Fehler oder an einem Fehler einer verwendeten Komponente?

C Dashboard-Vorlage der Rolle des Projektmanagers innerhalb des SSE Labs

Tabelle C.1: Dashboard-Vorlage der Rolle des Projektmanagers im SSE Lab

Qualitätsmerkmal	Fragen	Metriken
PROJEKTQUALITÄT		
Planungssicherheit		
<i>Zeit</i>	<ul style="list-style-type: none"> • Verläuft das Projekt nach Plan? • Können Projekttermine eingehalten werden? 	<ul style="list-style-type: none"> • Projekt-Fortschritt (in %) [anhand Auswertung des Ticketsystems] • Meilensteinplan [anhand Auswertung des Ticketsystems]
Projektleistung		
<i>Fortschritt</i>	<ul style="list-style-type: none"> • Wie verläuft die Ticketabarbeitung innerhalb des Projekts? • Wie lange dauert die durchschnittliche Ticketabarbeitung? • Wie verläuft die Abarbeitung offener Fehler? 	<ul style="list-style-type: none"> • Historie Anzahl abgearbeiteter Tickets [Erhebungsrahmen: täglich, wöchentlich, monatlich, im Quartal, jährlich] • Durchschnittliche Ticketdauer (in Stunden oder Minuten) • Historie Anzahl abgearbeiteter Fehler [Erhebungsrahmen: täglich, wöchentlich, monatlich, im Quartal, jährlich]

Qualitätsmerkmal	Fragen	Metriken
<i>Qualität</i>	<ul style="list-style-type: none">• Wie entwickeln sich Fehler innerhalb der Projektlaufzeit?• Wie hat sich die Anzahl der Tests innerhalb des Projekts entwickelt?• Wie verläuft die Entwicklung des Testerfolgs?• Wie entwickelt sich die Testabdeckung?• Wie verlaufen die Builds im Allgemeinen?	<ul style="list-style-type: none">• Historie Fehlerentwicklung (offene, geschlossene, kritische Fehler) [Erhebungsrahmen: wöchentlich, monatlich, im Quartal]• Historie Testanzahl (ggf. zusätzlich im Vergleich zu Codeentwicklung)• Historie Testerfolg (in %) [Erhebungsrahmen: nach Commits, täglich, wöchentlich, monatlich, im Quartal]• Historie Testabdeckung [Erhebungsrahmen: nach Commits, täglich, wöchentlich, monatlich, im Quartal]• Historie Buildverlauf [Erhebungsrahmen: nach Commits, täglich, wöchentlich, monatlich, im Quartal]

Qualitäts- merkmal	Fragen	Metriken
-----------------------	--------	----------

PRODUKTQUALITÄT

Prüfbarkeit

Testabdeckung

- | | |
|---|---|
| <ul style="list-style-type: none">• Wie hoch ist der aktuelle Testerfolg?• Wie hat sich der Testerfolg nach dem letzten Commit verändert?• Wie hoch ist die momentane Testabdeckung?• Welcher Commit hat die Testabdeckung verschlechtert? | <ul style="list-style-type: none">• Aktueller Testerfolg (in %)• Trend Testerfolg nach Commit (bei Verschlechterung mit Commit-Nummer)• Testabdeckung (in %)• Trend Testabdeckung nach Commit (bei Verschlechterung mit Commit-Nummer) |
|---|---|

Reife

Fehler

- | | |
|---|---|
| <ul style="list-style-type: none">• Wie hoch ist die aktuelle Zahl der offenen Fehler?• Wie viele kritischen Fehler existieren im Projekt?• Wie verteilen sich die Fehler nach Härtegrad?• Wie sieht die Verteilung der Fehler nach Priorität aus?• Wie verteilen sich die Fehler bezüglich des Abarbeitungsstatus? | <ul style="list-style-type: none">• Aktuelle Anzahl offener Fehler• Anzahl kritischer Fehler• Anzahl der Fehler nach Härtegrad• Anzahl der Fehler nach Priorität• Anzahl der Fehler nach Abarbeitungsstatus |
|---|---|
-

Qualitätsmerkmal	Fragen	Metriken
Änderbarkeit		
<i>Dokumentation</i>		
	<ul style="list-style-type: none">• Wie hoch ist die Dichte der Kommentare innerhalb des Quellcodes?• Besitzt jede Klasse/Methode einen Kommentar?• Wie hoch ist der Verletzungsgrad der Kodierrichtlinien?• Welches sind die Top-5-Regelverletzungen innerhalb dieser Richtlinien?	<ul style="list-style-type: none">• Kommentardichte (in %)• Anzahl der Klassen ohne Kommentare• Anzahl der Methoden ohne Kommentare• Anzahl der Verletzungen der Kodierrichtlinien (u.a. auch Verteilung nach Härtegrad)• Top-5-Regelverletzungen der Kodierrichtlinien
Zuverlässigkeit		
<i>Build</i>		
	<ul style="list-style-type: none">• Wie verläuft der Build?• Welche Probleme treten beim Build auf?• Welcher Commit hat Probleme in Build verursacht?	<ul style="list-style-type: none">• Trend aktueller Buildstatus• Dauer aktueller Build• Builderfolg nach Commit (bei Misserfolg mit Commit-Nummer und/oder Fehlermeldung)

D Visio-Konfigurationsprototyp des SSELab-Dashboards

Folglich wird eine Auswahl des mit *Microsoft Visio* erstellten Konfigurationsprototypen für das SSELab dargestellt. Innerhalb des Feedbacks wurde klar, dass dieser eine große Überarbeitung bedarf, woraufhin ein weiterer Papier-Prototyp zur Konfigurationsansicht erstellt wurde.

Folgende Makel wurden aufgedeckt:

- Zu viele Informationen innerhalb einer Ansicht (Abbildung D.2): Die Informationsbedürfnisse sollten nach den Merkmalen des Qualitätsmodells gruppiert und schnell überblickbar sein
- Verbindungen beziehungsweise Zugehörigkeiten zwischen Fragestellungen und Widgets sollten durch Highlighting ersichtlich werden
- Möglichkeit der Suche nach Stichwörtern (Qualitätsmerkmal, Fragestellung, Widget) innerhalb der Konfiguration für schnelle Auswahl einfügen
- Vorgehen der Konfiguration sollte sowohl für unerfahrene Benutzer, als auch Experten angelegt sein: entweder Konfiguration mit Hilfestellungen bezüglich der Informationsbedürfnisse oder Möglichkeit der Direktauswahl von Metriken
- Markierung der Dashboard-Widgets nach Detaillierungsgrad: Einige Widgets enthalten Informationen mehrerer Fragestellungen. Durch eine eindeutige Markierung wird dieser Detaillierungsgrad angezeigt, so dass die Möglichkeit der Auswahl weniger Widgets hohen Detailgrad oder vieler Widgets mit wenigen Details existiert

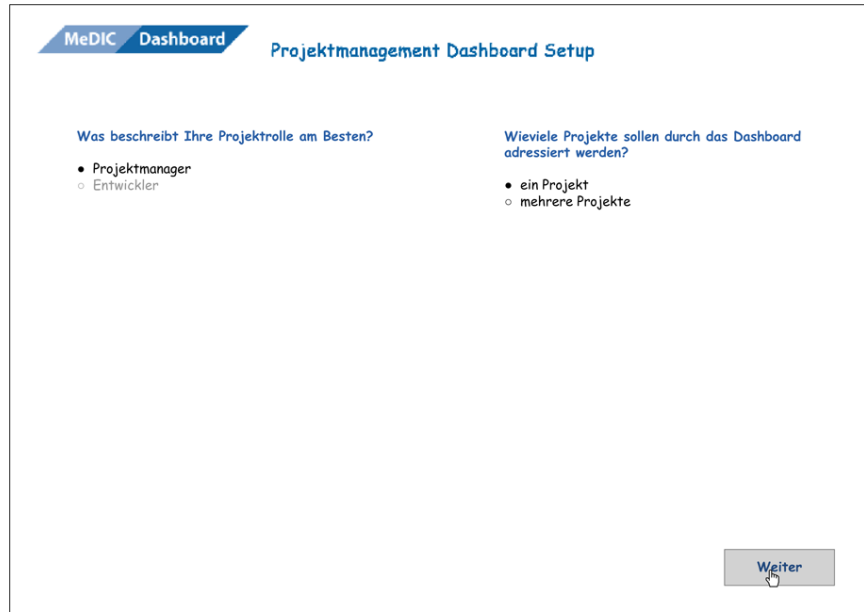


Abbildung D.1: Schritt 1: Auswahldialog zur Wahl der Benutzerrolle

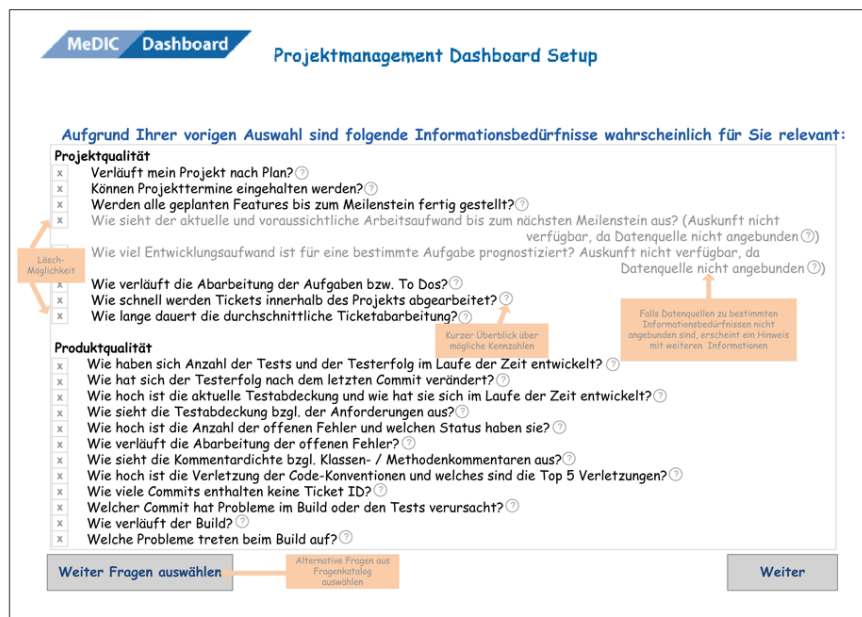


Abbildung D.2: Schritt 2: Konfiguration der Informationsbedürfnisse

MeDIC Dashboard **Projektmanagement Dashboard Setup**

Aufgrund Ihrer vorigen Auswahl sind folgende Informationsbedürfnisse wahrscheinlich für Sie relevant:

Informationsbedürfnisse

Projektqualität

- Verläuft mein Projekt nach Plan?
- Können Projekttermine eingehalten werden?
- Wenden alle geplanten Features bis zum Meilenstein fertig gestellt?
- Wie verläuft die Abarbeitung der Aufgaben bzw. To Dos?
- Wie schnell werden Tickets innerhalb des Projekts abgearbeitet?
- Wie lange dauert die durchschnittliche Ticketbearbeitung?

Produktqualität

- Wie haben sich Anzahl der Tests und der Testerfolg im Laufe der Entwicklung verändert?
- Wie hoch ist der Testerfolg nach dem letzten Commit verändert?
- Wie hoch ist die aktuelle Testabdeckung und wie hat sie sich im Laufe der Zeit entwickelt?
- Wie sieht die Testabdeckung bzgl. der Anforderungen aus?
- Wie hoch ist die Anzahl der offenen Fehler und welchen Status haben sie?**
- Wie verläuft die Abarbeitung der offenen Fehler?
- Wie sieht die Kommentardichte bzgl. Klassen- / Methodenkommentaren aus?
- Wie hoch ist die Verletzung der Code-Konventionen und welches sind die Top 5 Verletzungen?
- Wie viele Commits enthalten keine Ticket ID?
- Welcher Commit hat Probleme im Build oder den Tests verursacht?
- Welche Probleme treten beim Build auf?

Dashboard-Widgets

Projektqualität

- Aktuelle Projekt Kennzahlen
- Anzahl offener Tickets
- Anzahl Commits ohne ID
- Anzahl offener Fehler
- Testerfolg
- Testabdeckung
- Meilensteinplan
- Fortschrittsanalyse
- Ticketevaluation (Anzahl, Status)
- Ticketbearbeitung (Dauer/ Aufwand)

Produktqualität

- Fehleranalyse (Historie aufgrund von Ticketdaten)
- Offene Fehler (Anzahl, Status)
- Testerfolg
- Testabdeckung
- Kommentardichte (nach Klassen)
- Regelverletzungen Coding (Anzahl, Top 5)
- Commitanalyse (Anzahl, ohne ID)
- Build-Status

Das Widget Aktuelle Projekt Kennzahlen beinhaltet mehrere Kennzahlen

Überblick über entsprechende Kennzahl (inkl. Darstellung und Interpretationshilfe)

Widgets können aus der Auswahl gelöst werden und tauchen somit nicht in der Konfiguration auf

Weiter Fragen auswählen **Weiter**

Abbildung D.3: Schritt 3: Konfiguration der Dashboard-Widgets

Aktuelle Projekt Kennzahlen

- Anzahl offene Fehler
- Anzahl offene Tickets
- Anzahl Commits ohne ID
- Testerfolg (in %)
- Testabdeckung (in %)

Mülleimer

Verwurf des Widgets durch Drag & Drop in den Mülleimer

Ziehen Sie ein Dashboard-Element auf den Mülleimer um es zu löschen

Metrik-Warenkorb

[Projekt] Aktuelle Projekt Kennzahlen

- [F] Verläuft mein Projekt nach Plan?
- [A] buildErfolgAPK

[Projekt] Meilensteinplan

- [F] Wie sieht der Projektplan aus?
- [A] meilensteinplan

[Produkt] Offene Fehler (Anzahl, Status)

- [F] Wie hoch ist die Anzahl der offenen Fehler?
- [F] Welchen Status haben die offenen Fehler?
- [A] chartOffeneFehler

[Produkt] Fehleranalyse (Historie)

- [F] Welchen Status haben die aktuellen Fehler?
- [F] Wie entwickelt sich die Abarbeitung der Fehler?
- [F] Wie sieht die Verteilung der Fehler nach Status aus?
- [A] chartFehlerHistorie

[Produkt] Build-Status

- [F] Wie verläuft der Build?
- [F] Welche Probleme treten beim Build auf?
- [F] Welcher Commit ist für neue Buildprobleme verantwortlich?
- [A] chartAnforderungsPhase

Bei Fertigstellung Dashboard sperren

MeDIC Dashboard **Projektmanagement Dashboard**

X Logout Dashboardübersicht Report Erzeugen Hilfe

Dashboard sperren

Abbildung D.4: Schritt 4: Konfiguration des Dashboards

E Ziele und Fragen der GDIS-Anforderungsanalyse

Informationsbedürfnisse, die anhand der Anforderungsanalyse durch SSELab-Projektleiter erhoben wurden, werden folglich anhand von Fragestellungen dieser dargestellt.

Die Klammern am Ende einer Frage beinhaltet die Zuordnung der entsprechenden AGSM-Stufen. Existiert ein * hinter der Klammer, so besteht eine Abhängigkeit zu der vorigen Frage. Wird die Markierung (**SE**) oder (**agil**) hinzugefügt, so ist diese Frage nur für SE-Projekte bzw. agile Projekte interessant. Weiterhin gilt folgende Definition:

[A] Frage, welche für allen Stufen relevant ist

[Konz] Frage mit Relevanz in der Phase der Konzeption

[Entw] Frage mit Relevanz in der Phase des Entwurfs

[Real] Frage mit Relevanz in der Phase der Realisierung

[Einf] Frage mit Relevanz in der Phase der Einführung

E.1 Prozessqualität

E.1.1 Planungssicherheit

E.1.1.1 Termineinhaltung

- Verläuft das Projekt nach Plan? [A]
 - Wie hoch ist die Planabweichung? [A]
 - Welcher Mehraufwand entsteht, wenn der Plan trotz Verzug eingehalten werden soll? [A]*
 - Wie wird sich das Projekt voraussichtlich entwickeln? [A]
- Kann der nächste Meilenstein eingehalten werden? [A]
- Wurde das Projekt im Plan abgeschlossen? [Einf]

E.1.1.2 Aufwandseinhaltung

Budget

- Liegt das Budget im Plan? [A]
- Welchen entspricht das aktuelle Leistungsmaß (*Earned Value*)? [A]
- Wie verteilt sich das Budget auf IT- und Fach-PT? [A]
- Wie viel Budget haben die einzelnen Lieferanten zur Verfügung? [A]
- Liegt das Budget der Lieferanten innerhalb des Planwerts? [A]
- Wie viele der zur Verfügung gestellten PT wurden tatsächlich verbraucht? [Einf]
- Wie hoch war die Planabweichung des Projekts, beziehungsweise welcher Mehraufwand ist entstanden? [Einf]

Verfügbarkeiten

- Wie sieht der aktuelle Staffingplan aus? [A]
- Sind die gestafften Mitarbeiter verfügbar? [A]
 - Wie sieht die Mitarbeiter-Verfügbarkeit der nächsten Wochen aus? [A]
 - Gibt es Mitarbeiter-Engpässe? [A]*

E.1.1.3 Lieferantentreue

- Welchen Status haben die Ergebnisse der im Projekt eingebundenen Lieferanten? [A]
- Bei wie vielen Lieferanten steht die Lieferung noch aus? [A]
- Liegen die Produktlieferungen der Lieferanten im Plan?[A]
- Existieren neue Lieferaufträge? [A] (SE)

E.1.2 Projektleistung

E.1.2.1 Prozesstransparenz

- Wie verläuft die Abarbeitung der offenen Punkte? [A]
 - Wie hoch ist die Anzahl der offenen Punkte? [A]
 - Welchen Status haben die offenen Punkte? [A]
 - Welche offenen Punkte müssen bis zum nächsten Meilenstein bearbeitet werden? [A]

- Wie läuft die Abarbeitung der Non-Compliance? [A]
 - hoch ist die Anzahl der Non-Compliance? [A]
 - Welchen Status haben die aktuellen Non-Compliance? [A]
 - Welche Non-Compliance müssen bis zum nächsten Meilenstein gelöst werden? [A]
- Wie verläuft die Abarbeitung der Prozess-Reviews aus? [A]
 - Wie viele Prozess-Reviews sind aktuell noch zu bearbeiten? [A]
 - Welchen Status haben die vorhandenen Prozess-Reviews? [A]
 - Welche Prozess-Reviews müssen bis zum nächsten Meilenstein durchgeführt werden? [A]
- Welche Projektrisiken existieren? [Konz, Entw]
 - Wie sind die ermittelten Risiken gewichtet? [Konz, Entw]
 - Wie hoch ist die Eintrittswahrscheinlichkeit der ermittelten Risiken? [Konz, Entw]
 - Welche Gegenmaßnahmen zu den Risiken existieren? [A]
 - Welche Risiken sind eingetreten? [A]

E.1.2.2 Entwicklungsgeschwindigkeit

- Wie entwickelt sich das Budget? [A]
- Wird der Mastertestplan eingehalten? [A]
 - Wie verläuft die Abarbeitung der Testfälle? [A]
 - Wie verläuft die Testdurchführung der einzelnen Teststufen? [Konz,Real]
- Wie verläuft die Fehlerabarbeitung? [A]
 - Wie sieht der Verlauf der kritischen Fehler aus? [A]
- Wie verläuft die Review-Abarbeitung? [A]
- Wie sieht der Verlauf der Anforderungsabarbeitungen aus? [A]
- Wie verläuft die Abarbeitung der Change Requests? [A]

E.2 Produktqualität

E.2.1 Prüfbarkeit

E.2.1.1 Testbarkeit

- Welchen Status haben die vorhandenen Tests? [A]

- Wie hoch ist die Anzahl der fehlschlagenden Tests? [A]
- Wie viele Tests müssen überarbeitet werden? [A]
- Welchen Status haben die einzelnen Teststufen? [Real, Einf]
 - Wie viele Test sind noch zum Abschluss einer Teststufe erfolgreich durchzuführen? [Real, Einf]
 - Welche Teststufen wurden abgeschlossen? [Einf]

E.2.2 Zuverlässigkeit

E.2.2.1 Korrektheit

Baseline

- Haben sich Anforderungen der Baseline geändert? [A]
- Welchen Status haben die Change Requests? [A]
- Wurden alle Change Requests ordnungsgemäß abgearbeitet? [A]

Anforderungen

- Welchen Status haben die vorhandenen Anforderungen? [A]
 - Sind neue Anforderungen hinzugekommen? [A]
 - Welche Prioritätensetzung der Anforderungen existieren? [A]
 - In welchen Phasen ist die Abarbeitung der einzelnen Anforderungen geplant? [A]
 - Welche Anforderungen müssen bis zum nächsten Meilenstein realisiert werden? [A]
 - Welche Abhängigkeiten existieren zwischen den Anforderungen? [Konz, Entw]
 - Sind alle Anforderungen durch Reviews geprüft worden? [A]
 - Wie hoch ist die Zahl der Reviews, die bis zur Fertigstellung der Phase noch durchgeführt werden müssen? [A]
- Wie hoch ist die Zahl der Anforderungen, die für den Testentwurf bereit stehen? [Konz, Entw, Real]
- Wie viele Anforderungen werden durch Testfälle abgedeckt? [Konz, Entw, Real]
- Wie verläuft die Abarbeitung des Product-Backlogs? [A] (agil)
- Welche Backlog-Items müssen in diesem Sprint realisiert werden? [A] (agil)
- Welche Änderungen des Backlogs haben sich ergeben? [A] (agil)

Reviews

- Welchen Status haben die vorhandenen Reviews? [A]
 - Wie hoch ist die Zahl der offene Reviews? [A]
 - Wie viele Reviews sind neu hinzugekommen? [A]

E.2.2.2 Reife

Fehler

- Welchen Status haben die aktuellen Fehler? [A]
 - Wie viele kritische Fehler existieren? [A]
 - Sind neue Fehler hinzu gekommen? [A]
- Wie verläuft die Fehlerabarbeitung? [A]
- Wie Dauer hat die Abarbeitung der Fehler? [A]

E.2.2.3 Betriebsabnahme

- Welchen Status haben die vorhandenen Abnahmekriterien? [Einf]
- Welche Ergebnisse liefert der Betriebstest? [Einf]
- Läuft das System stabil? [Einf]

E.3 Zusätzliche Fragen

- Welche wichtigen Termine stehen aktuell im Unternehmen an? [A]
- Welche wissenswerten Veränderungen sind passiert? [A]
- Welche wichtigen Ankündigungen wurden (innerhalb des Unternehmens) gemacht? [A]

F Dashboard-Vorlage der Rolle des Projektleiters innerhalb der GDIS GmbH

Tabelle F.1: Dashboard-Vorlage der Rolle des Projektleiters in der GDIS GmbH

Qualitätsmerkmal	Fragen	Metriken
PROJEKTQUALITÄT		
Planungssicherheit		
<i>Zeit</i>	<ul style="list-style-type: none"> • Verläuft das Projekt nach Plan? • Wie wird sich das Projekt voraussichtlich entwickeln? • Kann der nächste Meilenstein eingehalten werden? • Welche offenen Punkte müssen bis zum nächsten Meilenstein bearbeitet werden? • Welche Non-Compliance müssen bis zum nächsten Meilenstein gelöst werden? • Welche Prozess-Reviews müssen bis zum nächsten Meilenstein durchgeführt werden? • Können Projekttermine eingehalten werden? • Liegen die Produktlieferungen der Lieferanten im Plan? 	<ul style="list-style-type: none"> • Projekt-Fortschritt (in %) [inkl. Vorhersage] • Meilensteinplan • Termineinhaltung [Termine inkl. Aufgaben (offene Punkte, Non-Compliance, Reviews)] • Liefertantenplan [Angaben zu Lieferanten, Zeitplan, Status]

Qualitätsmerkmal	Fragen	Metriken
<i>Aufwand</i>	<ul style="list-style-type: none"> • Liegt das Budget im Plan? • Wie hoch ist die Planabweichung? • Wie entwickelt sich das Leistungsmaß (Earned Value)? • Wie sieht die Budgetverteilung zwischen IT- und Fach-PT aus? • Wie viele zur Verfügung gestellten PF wurden tatsächlich verbraucht? • Liegt das Budget der Lieferanten innerhalb des Planwerts? 	<ul style="list-style-type: none"> • Aufwandsanalyse (in PT) [inkl. Verbrauch, Vorhersage, Planabweichung; Erhebungsrahmen: wöchentlich, monatlich, im Quartal, jährlich] • Earned-Value-Analyse [Erhebungsrahmen: wöchentlich, monatlich, im Quartal, jährlich] • Verteilung von IT- und Fach-PT (in PT) [Erhebungsrahmen: wöchentlich, monatlich, im Quartal, jährlich] • Analyse Liferantenbudget (in PT) [Verteilung auf unterschiedliche Lieferanten, Planabweichung; [Erhebungsrahmen: wöchentlich, monatlich, im Quartal, jährlich]]
<i>Personal</i>	<ul style="list-style-type: none"> • Wie sieht der aktuelle Staffingplan aus? • Wie sieht die Mitarbeiter-Verfügbarkeit der nahen Zukunft aus? • Gibt es Mitarbeiter-Engpässe? 	<ul style="list-style-type: none"> • Auflistung gestaffter Mitarbeiter [inkl. Aufgabentätigkeit] • Staffingplan (mit Ausblick der kommenden 2 Arbeitswochen) • Engpass-Schätzung [innerhalb der kommenden 2-4 Arbeitswochen]

Qualitäts- merkmal	Fragen	Metriken
-----------------------	--------	----------

Projektleistung

Fortschritt

- | | |
|---|---|
| <ul style="list-style-type: none">• Wie verläuft die Abarbeitung der Testfälle?• Wie verläuft die Abarbeitung der offenen Fehler?• Wie verläuft die Abarbeitung der offenen Reviews?• Wie entwickelt sich die Abarbeitung der Anforderungen?• Wie sieht der Verlauf der Change Requests und dessen Abarbeitung aus? | <ul style="list-style-type: none">• Historie Anzahl abgearbeiteter Testfälle [Erhebungsrahmen: täglich, wöchentlich, monatlich, im Quartal, jährlich]• Historie Anzahl abgearbeiteter und neuer Fehler [Erhebungsrahmen: täglich, wöchentlich, monatlich, im Quartal, jährlich]• Historie Anzahl abgearbeiteter und neuer Reviews [Erhebungsrahmen: wöchentlich, monatlich, im Quartal, jährlich]• Historie Anzahl abgearbeiteter und neuer Anforderungen [Erhebungsrahmen: wöchentlich, monatlich, im Quartal, jährlich]• Historie Anzahl abgearbeiteter und neuer Change Requests [Erhebungsrahmen: wöchentlich, monatlich, im Quartal, jährlich] |
|---|---|
-

Qualitäts- merkmal	Fragen	Metriken
<i>Qualität</i>	<ul style="list-style-type: none"> • Wie hoch ist die Planabweichung? • Welchen Status haben existierende offene Punkte? • Welchen Status haben aktuelle Non-Compliance? • Welchen Status haben vorhandene Reviews? • Wie entwickeln sich Fehler innerhalb der Projektlaufzeit? • Wie hat sich die Anzahl der Tests innerhalb des Projekts entwickelt? • Wie verläuft die Entwicklung des Testerfolgs? • Wie entwickelt sich die Testabdeckung? • Welche Projektrisiken existieren und wie hoch ist ihre Eintrittswahrscheinlichkeit? • Wie sind die Projektrisiken gewichtet? 	<ul style="list-style-type: none"> • Planabweichung (in % oder PT) • Status offene Punkte [Anzahl, Status] • Status Non-Compliance [Anzahl, Abarbeitungsstatus] • Status Reviews [Anzahl, Status] • Historie Fehlerentwicklung (offene, geschlossene, kritische Fehler) [Erhebungsrahmen: täglich, wöchentlich, monatlich, im Quartal] • Historie Testanzahl (ggf. zusätzlich im Vergleich zu Codeentwicklung) • Historie Testerfolg (in %) [Erhebungsrahmen: täglich, wöchentlich, monatlich, im Quartal] • Historie Testabdeckung [Erhebungsrahmen: täglich, wöchentlich, monatlich, im Quartal] • Risikoanalyse [Anzahl, Gewichtung, Eintrittswahrscheinlichkeit]

Qualitäts- merkmal	Fragen	Metriken
-----------------------	--------	----------

PRODUKTQUALITÄT

Prüfbarkeit

Testabdeckung

- | | |
|--|--|
| <ul style="list-style-type: none">• Wie hoch ist der aktuelle Testerfolg?• Wie viele Tests schlagen fehl?• Wie viele Tests müssen überarbeitet werden? | <ul style="list-style-type: none">• Aktueller Testerfolg (in %)• Anzahl an Testfehlschlägen• Anzahl an zu überarbeitende Tests• Teststufenanalyse [Anzahl, Status, Verlauf] |
|--|--|

Zuverlässigkeit

Anforderungen

- | | |
|--|---|
| <ul style="list-style-type: none">• Haben sich neue Anforderungen bezüglich der Baseline ergeben?• Welchen Status haben die Change Requests?• Welchen Status und welche Priorität haben die vorhandenen Anforderungen? | <ul style="list-style-type: none">• Anforderungsanalyse [Anzahl, Status, Priorität, Testabdeckung]• Change-Request-Analyse [Anzahl, Status, Priorität] |
|--|---|

Reviews

- | | |
|---|---|
| <ul style="list-style-type: none">• Welchen Status und welche Priorität haben vorhandene Reviews? | <ul style="list-style-type: none">• Reviewanalyse [Anzahl, Status, Priorität] |
|---|---|
-

Qualitätsmerkmal	Fragen	Metriken
<i>Fehler</i>	<ul style="list-style-type: none">• Wie hoch ist die aktuelle Zahl der offenen Fehler?• Wie viele kritischen Fehler existieren im Projekt?• Wie verteilen sich die Fehler nach Härtegrad?• Wie sieht die Verteilung der Fehler nach Priorität aus?• Wie verteilen sich die Fehler bezüglich des Abarbeitungsstatus?• Wie lange dauert die durchschnittliche Fehlerabarbeitung?	<ul style="list-style-type: none">• Fehleranalyse [Anzahl offener und kritischer Fehler, Härtegrad, Priorität, Abarbeitungsstatus]• Durchschnittliche Abarbeitungsdauer offener Fehler (in Stunden oder Tagen)

G Ziele und Fragen der i-nex-Anforderungsanalyse

Informationsbedürfnisse, die anhand der Anforderungsanalyse bei der i-nex GmbH erhoben wurden, werden folglich anhand von Fragestellungen dargestellt. Sie sind nach dem dazugehörigen Qualitätsmodell (siehe Abbildung V.2.1) in Informationsbedürfnisse bzgl. der Projekt- und Produktqualität aufgeteilt.

Fragen, welche mit [Z] markiert wurden, können zum aktuellen Zeitpunkt ohne weiteren Anpassungsaufwand der Mess-Tools bzw. der Entwicklungsprozesse nicht direkt integriert werden, wurden jedoch auch als wichtig bzw. interessant identifiziert.

G.1 Projektqualität

G.1.1 Planungssicherheit

- Verläuft das Projekt nach Plan?
- Können Projekttermine eingehalten werden?
- Wie sieht der aktuelle und voraussichtliche Arbeitsaufwand bis zum nächsten Meilenstein aus?
- Werden alle geplanten Features bis zum Release fertig gestellt?
- Werden alle offenen Tickets bzw. Arbeitspakete bis zum Release abgearbeitet?
- Kann das nächste Release planmäßig stattfinden?
- Wie entwickeln sich die Projektkosten?
- Liegen Kosten und Aufwand im Plan?
- Wie hoch ist der Aufwand der Nachbearbeitung?
- Welche Kapazitäten sind vorhanden?
- Existieren Kapazitätsengpässe?
- Wie sind die Ressourcen in den Projekten verteilt?
- Wie gut verläuft der Entwicklungsprozess? [Z]

G.1.2 Projektleistung

- Wie verläuft die Ticketabarbeitung innerhalb des Projekts?
- Wie lange dauert die durchschnittliche Ticketabarbeitung?
- Werden hoch-priorisierte Tickets zügig abgearbeitet?
- Wie viel Aufwand hat die Abarbeitung der Aufgaben/ Tickets tatsächlich gekostet?
- Wie schnell werden interne und externe Projekte abgearbeitet?

G.1.3 Kundenzufriedenheit [Z]

- Ist der Kunde mit dem Projektverlauf zufrieden? [Z]
- Entspricht die Kommunikation den Vorstellungen des Kunden? [Z]
- Ist der Kunde mit dem Ergebnis zufrieden? [Z]

G.2 Produktqualität

G.2.1 Prüfbarkeit

- Wie verlaufen die einzelnen Testphasen? [Z]
- Wie sieht der aktuelle Testerfolg des Produkts aus? [Z]
- Wie hat sich der Testerfolg innerhalb des Projekts entwickelt? [Z]
- Wie hat sich die Anzahl der Tests innerhalb des Projekts entwickelt? [Z]
- Wie hat sich der Testerfolg nach dem letzten Commit verändert? [Z]
- Wie verläuft der Integrationstest? [Z]
- Wie hoch ist die aktuelle Testabdeckung? [Z]
- Wie hat sich die Testabdeckung innerhalb der Projekt-Laufzeit verändert? [Z]
- Existieren Codeduplikate und wenn ja, wie viele?
- Welche CSS-Dateien werden innerhalb der Projekte nicht mehr genutzt?
- Welche JSF-Templates werden nicht mehr verwendet?

G.2.2 Änderbarkeit

- Wie hoch ist die Kopplung von Klassen und Paketen innerhalb des Projekts? [Z]
- Wie hoch ist die Modularität der Komponenten? [Z]
- Wie sieht die Rate der Wiederverwendbarkeit der vorhandenen Artefakte aus?
- Wie gut ist die vorhandene Architektur? [Z]
- Wie hoch ist die Dokumentationsdichte im Quellcode?

G.2.3 Reife

- Wie sieht die aktuelle Anzahl der offenen Fehler aus?
- Wie sieht der Fehlerstatus bzw. die Verteilung nach Härtegrad der Fehler aus?
- Wie verläuft die Abarbeitung der offenen Fehler?
- Wie sieht die Entwicklung der Fehler innerhalb der Projekt-Laufzeit aus?
- Wie viele Fehler treten nach dem Release auf?
- Welche typischen Fehler treten nach dem Release auf?

G.2.4 Zuverlässigkeit

- Wie verläuft der Build?
- Welche Probleme treten beim Build auf?
- Welcher Commit hat Probleme beim Build verursacht?
- Welcher Commit hat den Testerfolg bzw. die Testabdeckung verschlechtert? [Z]
- Wie entwickelt sich die Projektqualität aufgrund der (letzten) Commits?
- Wie entwickelt sich die Speichernutzung?
- Wie entwickelt sich die Laufzeit?

H Dashboard-Vorlage der Rolle des Geschäftsführers innerhalb der i-nex GmbH

Tabelle H.1: Dashboard-Vorlage der Rolle des Projektleiters im der i-nex GmbH

Qualitätsmerkmal	Fragen	Metriken
PROJEKTQUALITÄT		
Planungssicherheit		
<i>Zeit</i>		
	<ul style="list-style-type: none"> • Verläuft das Projekt nach Plan? • Können Projekttermine eingehalten werden? • Werden alle geplanten Features bis zum Release fertig gestellt? • Werden alle offenen Tickets bzw. Arbeitspakete bis zum Release abgearbeitet? • Kann das nächste Release planmäßig stattfinden? 	<ul style="list-style-type: none"> • Projekt-Fortschritt (in %) • Meilensteinplan • Anzahl offener Tickets bis zum nächsten Release [inkl. Aufwandsprognose]
<i>Personal</i>		
	<ul style="list-style-type: none"> • Welche Kapazitäten sind vorhanden? • Existieren Kapazitätsengpässe? • Wie sind die Ressourcen in den Projekten verteilt? 	<ul style="list-style-type: none"> • Auflistung der verfügbaren Mitarbeiter [inkl. Aufgabentätigkeit, Projektzuständigkeiten und Urlaubsplan] • Engpass-Schätzung [innerhalb der kommenden 2-4 Arbeitswochen]

Qualitätsmerkmal	Fragen	Metriken
Projektleistung		
<i>Fortschritt</i>		
	<ul style="list-style-type: none">• Wie verläuft die Ticketabarbeitung innerhalb des Projekts?• Wie lange dauert die durchschnittliche Ticketabarbeitung?• Wie schnell werden interne und externe Projekte abgearbeitet?	<ul style="list-style-type: none">• Historie Anzahl abgearbeiteter Tickets [Erhebungsrahmen: täglich, wöchentlich, monatlich, im Quartal, jährlich]• Durchschnittliche Ticketdauer (in Stunden oder Minuten)• Historie Projektvortschritt (intern vs. extern in %)

PRODUKTQUALITÄT

Reife

Fehler

- | | |
|--|---|
| <ul style="list-style-type: none">• Wie sieht die aktuelle Anzahl der offenen Fehler aus?• Wie sieht der Fehlerstatus bzw. die Verteilung nach Härtegrad der Fehler aus?• Wie verläuft die Abarbeitung der offenen Fehler?• Wie sieht die Entwicklung der Fehler innerhalb der Projekt-Laufzeit aus?• Wie viele Fehler treten nach dem Release auf?• Welche typischen Fehler treten nach dem Release auf? | <ul style="list-style-type: none">• Aktuelle Anzahl offener Fehler• Historie Anzahl Fehler (nach Status) [Erhebungsrahmen: täglich, wöchentlich, monatlich, im Quartal, jährlich]• Anzahl kritischer Fehler• Anzahl der Fehler nach Härtegrad• Anzahl der Fehler nach Priorität• Anzahl der Fehler nach Release• Auflistung typischer Fehler nach Release |
|--|---|
-

Literaturverzeichnis

- [BCR94] BASILI, Victor R. ; CALDIERA, Gianluigi ; ROMBACH, H. D.: *The Goal Question Metric Approach: Encyclopedia of Software Engineering*. 1994
- [BG08] BERGMANN, Rainer ; GARRECHT, Martin: *Organisation und Projektmanagement*. Heidelberg : Physica-Verlag Heidelberg, 2008 (BA KOMPAKT). – ISBN 9783790820188
- [BMW⁺00] BOURNE, Mike ; MILLS, John ; WILCOX, Mark ; NEELY, Andy ; PLATTS, Ken: Designing, implementing and updating performance measurement systems. In: *International Journal of Operations & Production Management* 20 (2000), Nr. 7, 754-771. http://www.vigilwiki.com/Content/PDF_Documents/Performance_Measurement/General/Designing_Implementing_and_Updating_Performance_Measurement_Systems.pdf
- [BR04] BENNICKE, Marcel ; RUST, Heinrich ; VIRTUELLES SOFTWARE ENGINEERING KOMPETENZZENTRUM (Hrsg.): *Messen im Software-Engineering und metrikbasierte Qualitätsanalyse*. Version: Report: ViSEK/024/D, 19.01.2004. http://www.software-kompetenz.de/servlet/is/15797/ViSEK_Report_024D_1.2.pdf?command=downloadContent&filename=ViSEK_Report_024D_1.2.pdf. (Version 1.2)
- [Cho11] CHOLAKOV, Stefan: *Conception of Collaborative Project Cockpits with Integrated Interpretation Aids*. Aachen, RWTH Aachen University, Diss., 03/2011. https://www2.swc.rwth-aachen.de/docs/2011_MA_Cholakov.pdf
- [Chr13] CHRISTIAN CHARLES: *Entwurf eines generischen Prozessleitstandes für Change Request Systeme*. Aachen, RWTH Aachen University, Diss., 06/2013. https://www2.swc.rwth-aachen.de/?post_type=thesis&p=945
- [Deu09] DEUTSCHES INSTITUT FÜR NORMIERUNG E.V.: *Projektmanagement - Projektmanagementsysteme - Teil 1: Grundlagen*. <http://www.beuth.de/de/norm/din-69901-1/113428320?SearchID=505180690>. Version: DIN 69901-1:2009-01, 01.2009
- [DH07] DREWS, Günter ; HILLEBRAND, Norbert: *Lexikon der Projektmanagement-Methoden*. 1. Freiburg [u.a.] : Haufe, 2007. – ISBN 3448080527

- [Eme12] EMELYANOVA, Elena: *Regelbasierte Initialisierung von Projektdashboards*. Aachen, RWTH Aachen University, Diss., 08/2012. https://www2.swc.rwth-aachen.de/?post_type=thesis&p=939
- [Eve12] EVERS, Frederic: *Konzeptionelle Erweiterung von Projektdashboards für unerfahrene Anwender*. Aachen, RWTH Aachen University, Diss., 05/2012. https://www2.swc.rwth-aachen.de/docs/MA_Evers.pdf
- [Fen91] FENTON, Norman E.: *Software metrics: A rigorous approach*. 1. London and New York : Chapman & Hall, 1991. – ISBN 9780412404405
- [Few04] FEW, Stephen: Dashboard Confusion. In: *Intelligent Enterprise* 03/2004 (2004). http://www.perceptualedge.com/articles/ie/dashboard_confusion.pdf
- [Few06] FEW, Stephen: *Information Dashboard Design: The Effective Visual Communication of Data*. 1. Beijing and Cambridge [MA] : O'Reilly, 2006. – ISBN 9780596100162
- [Few12] FEW, Stephen: *Show Me the Numbers: Designing Tables and Graphs to Enlighten*. 2. Burlingame and Calif : Analytics Press, 2012. – ISBN 9780970601971
- [Gen11] GENERALI DEUTSCHLAND SERVICES GMBH: *Konzernrichtlinie Projektmanagement Projektmanagementleitfaden*. 2011
- [Gor13] GORA, Matthias: *Entwicklung eines Dashboard Prototyping-Werkzeugs*. Aachen, RWTH Aachen University, Diss., 03/2013. https://www2.swc.rwth-aachen.de/docs/2013_BA_Gora.pdf
- [GS11] GLASS, Marco ; SCHMIDT, Thorsten: Aufbau und Weiterentwicklung der Datenqualitätsmessungen bei der Telekom Deutschland GmbH. In: *HMD – Praxis der Wirtschaftsinformatik* 48. Jahrgang (06/2011), Nr. 279, 56–63. http://www.pass-sales.com/fileadmin/media/redaktion_sales/downloads/fachartikel_HMD_279_Sonderdruck_Glass_PASS_IT_11_2011.pdf
- [HKR12] HERRMANN, Christoph ; KURPICK, Thomas ; RUMPE, Bernhard: SSELab: A plug-in-based framework for web-based project portals. In: *TOPI 2012*, IEEE, 2012. – ISBN 1467318205, S. 61–66
- [IBM03] IBM CORP.: *IBM Rational Unified Process: Version 2003.06.15*. 2003
- [Ins04] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS: *IEEE Guide Adoption of PMI Standard: A Guide to the Project Management Body of Knowledge*. IEEE Std. 1490-2003. New York and N.Y., 24.05.2004

-
- [Ins09] INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS: *IEEE standard adoption of ISO/IEC 15939:2007: Systems and software engineering–measurement process*. ISO/IEC 15939:2007. New York and N.Y., 30.01.2009
- [ISO01] ISO/IEC: *Software engineering – Product quality – Part 1: Quality model*. http://www.iso.org/iso/catalogue_detail.htm?csnumber=22749. Version: ISO/IEC 9126-1:2001(E), 15.06.2001
- [ISO10] ISO/IEC, IEEE: *Systems and software engineering–vocabulary*. http://www.iso.org/iso/catalogue_detail.htm?csnumber=50518. Version: ISO/IEC/IEEE 24765:2010(E), 15.12.2010
- [JP04] JOHNSON, Addie ; PROCTOR, Robert W.: *Attention: Theory and practice*. Thousand Oaks and Calif : Sage Publications, 2004. – ISBN 9780761927617
- [Kel07] KELTER, Udo: *Software-Qualitätsmodelle*. http://pi.informatik.uni-siegen.de/lehre/2012s/LM/lm_sqmo_20070505_a5.pdf. Version: 2007 (Vorlesungsmitschrift Softwaretechnik II)
- [KHL⁺11] KUSTER, Jürg ; HUBER, Eugen ; LIPPMANN, Robert ; SCHMID, Alphons ; SCHNEIDER, Emil ; WITSCHI, Urs ; WÜST, Roger: *Handbuch Projektmanagement*. 3. überarb. Berlin and Heidelberg : Springer-Verlag Berlin Heidelberg, 2011 (SpringerLink : Bücher). – ISBN 9783642212420
- [Kne06] KNEUPER, Ralf: *CMMI: Verbesserung von Softwareprozessen mit Capability Maturity Model Integration*. 2. Heidelberg : Dpunkt-Verl., 2006. – ISBN 9783898643733
- [Köh07] KÖHLER, Peter T.: *ITIL: Das IT-Servicemanagement Framework*. 2. Berlin and Heidelberg and New York : Springer, 2007 (Xpert.press). – ISBN 3540379509
- [Kru04] KRUCHTEN, Philippe: *The Rational Unified Process: An Introduction*. Addison-Wesley, 2004. – ISBN 9780321197702
- [LL07] LUDEWIG, Jochen ; LICHTER, Horst: *Software Engineering: Grundlagen, Menschen, Prozesse, Techniken*. 1. Heidelberg : Dpunkt-Verl., 2007. – ISBN 9783898642682
- [Mäd13] MÄDLER, Martin: *Variabilität von Metriken und Dashboard-Items im Umfeld von MeDIC*. Aachen, RWTH Aachen University, Diss., 06/2013. https://www2.swc.rwth-aachen.de/?post_type=thesis&p=943
- [NHL⁺08] NEUMANN, Karl J. ; HOLZMÜLLER, Günter ; LUCAS, Alexander ; PECHLI-VANIDIS, Stavros ; RAUTENBERG, Frank: *AGSM Kennzahlenmodell*. 2008

- [OB92] OIVO, M. ; BASILI, V.R: Representing software engineering models: the TA-ME goal oriented approach. In: *IEEE Transactions on Software Engineering* 18 (1992), Nr. 10, S. 886–898
- [Ott13] OTTO, Arthur: *Integration einer Metrik-Infrastruktur in die Projektverwaltung SSE-Lab*. Aachen, RWTH Aachen University, Diss., 2013. https://www2.swc.rwth-aachen.de/?post_type=thesis&p=1124
- [Pic08] PICHLER, Roman: *Srum - agiles Projektmanagement erfolgreich einsetzen*. 1. Heidelberg : Dpunkt-Verl., 2008. – ISBN 3898644782
- [PR98] PATZAK, Gerold ; RATTAY, Günter: *Projekt-Management: Leitfaden zum Management von Projekten, Projektportfolios und projektorientierten Unternehmen*. 3. Wien : Linde, 1998. – ISBN 9783851227574
- [Pre01] PRESSMAN, Roger S.: *Software engineering: A practitioner's approach*. 5. Boston and Mass : McGraw Hill, 2001 (McGraw-Hill series in computer science). – ISBN 9780071181822
- [PS13] PHAM, Raphael ; SCHNEIDER, Kurt: Smarter GQM-Editor mit verringerter Einstiegshürde. In: *Software Engineering im Unterricht der Hochschulen (SEUH 2013)*, 2013
- [Rat03] RATIONAL: *Rational Unified Process: Best Practices for Software Development Teams*. 10.01.2003 (Rational Software White Paper TP026B, Rev 11/01)
- [Sny03] SNYDER, Carolyn: *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. San Francisco and Calif : Morgan Kaufmann, Elsevier Science, 2003 (The Morgan Kaufmann series in interactive technologies). – ISBN 1558608702
- [Soe13] SOEST, Joris van: *Designing a performance dashboard at Sigmax*. Enschede, University of Twente, Diss., 05/2013. <http://essay.utwente.nl/63165/>
- [Som11] SOMMERVILLE, Ian: *Software engineering*. 9. Boston : Pearson, 2011. – ISBN 9780137053469
- [Ste13] STEFFENS, Andreas: *Entwurf eines Architekturmodells zur Integration heterogener Systeme in MeDIC*. Aachen, RWTH Aachen University, Diss., 02/2013. https://www2.swc.rwth-aachen.de/?post_type=thesis&p=850
- [Tok13] TOKAREV, Paul: *Entwicklung einer Komponente zum Export von Analyseergebnissen aus MeDIC Dashboard*. Aachen, RWTH Aachen University, Diss., 06/2013. https://www2.swc.rwth-aachen.de/?post_type=thesis&p=969

- [VLS] VIANDEN, Matthias ; LICHTER, Horst ; STEFFENS, Andreas: Towards a Maintainable Federalist Enterprise Measurement Infrastructure. In: *23rd International Workshop on Software Measurement (IWSM) and the 8th International Conference on Software Process and Product Measurement (Mensura)*
- [VMo12] *V-Modell XT Gesamt 1.4*. Version: 15.05.2012. http://v-modell.iabg.de/index.php?option=com_docman&task=doc_view&gid=48
- [WB11] WOLF, Henning ; BLEEK, Wolf-Gideon: *Agile Softwareentwicklung: Werte, Konzepte und Methoden*. 2. Heidelberg : dpunkt-Verl, 2011. – ISBN 3898647013
- [Wer12] WERNER, Florian: *Entwurf eines regelbasierten GQM-Editors mit geringer Einstiegshürde*. Hannover, Leibniz Universität Hannover, Diss., 09/2012. <http://www.se.uni-hannover.de/pub/File/pdfpapers/Werner2012.pdf>
- [WH00] WICKENS, Christopher D. ; HOLLANDS, Justin G.: *Engineering Psychology and Human Performance*. 3. Upper Saddle River and NJ : Prentice Hall, 2000. – ISBN 9780321013439
- [Wol12] WOLF, Henning: *Agile Projekte mit Scrum, XP und Kanban im Unternehmen durchführen: Erfahrungsberichte aus der Praxis*. dpunkt.verlag, 2012. – ISBN 3864910641

Internetquellen

- [1] *Git*. <http://git-scm.com/>, . – Zugriffsdatum: 11.06.2013
- [2] *Jenkins CI: An extendable open source continuous integration server*. <http://jenkins-ci.org/>, . – Zugriffsdatum: 31.05.2013
- [3] *JUnit: A programmer-oriented testing framework for Java*. <http://junit.org/>, . – Zugriffsdatum: 31.05.2013
- [4] *Sonar*. <http://www.sonarsource.org/>, . – Zugriffsdatum: 11.06.2013
- [5] AHB ELECTRONIC GMBH: *Zina - Zeiterfassung via Lotus Notes*. http://testdev.ahb-electronic.de/ahb/prod/pze_notes_zina.htm, . – Zugriffsdatum: 19.06.2013
- [6] APACHE SOFTWARE FOUNDATION: *Apache Subversion*. <http://subversion.apache.org/>, . – Zugriffsdatum: 11.06.2013
- [7] ATlassian: *JIRA: Issue & Project Tracking Software*. <http://www.atlassian.com/de/software/jira>, . – Zugriffsdatum: 31.05.2013
- [8] AXURE SOFTWARE SOLUTIONS, INC: *Axure RP*. <http://www.axure.com/features>, . – Zugriffsdatum: 17.06.2013
- [9] BALSAMIQ STUDIOS, LLC: *Balsamiq Mockups*. <http://www.balsamiq.com/products/mockups>, . – Zugriffsdatum: 17.06.2013
- [10] BIBLIOGRAPHISCHES INSTITUT GMBH: *Duden online: Qualität - Bedeutungen*. <http://www.duden.de/rechtschreibung/Qualitaet>, . – Zugriffsdatum: 20.04.2013
- [11] ECLIPSE FOUNDATION: *Eclipse*. <http://www.eclipse.org/>, . – Zugriffsdatum: 31.05.2013
- [12] ECLIPSE FOUNDATION: *Hudson Continuous Integration : Extensible continuous integration server*. <http://hudson-ci.org/>, . – Zugriffsdatum: 31.05.2013
- [13] EDGEWALL SOFTWARE: *The Trac Project*. <http://trac.edgewall.org/>, . – Zugriffsdatum: 31.05.2013
- [14] EIKIPEDIA: *Softwaremetrik*. <http://de.wikipedia.org/w/index.php?title=Softwaremetrik&oldid=117194898>, . – Zugriffsdatum: 03.06.2013

- [15] FREE SOFTWARE FOUNDATION INC: *Concurrent Versions System*. <http://savannah.nongnu.org/projects/cvs>, . – Zugriffsdatum: 19.06.2013
- [16] GENERALI DEUTSCHLAND SERVICES GMBH: *Generali Deutschland Informatik Services*. <http://www.generali-informatik.de/>, . – Zugriffsdatum: 02.04.2013
- [17] GENERALI DEUTSCHLAND SERVICES GMBH: *Generali Deutschland: Startseite*. <http://generali-deutschland.de/online/portal/gdinternet/de/content/>, . – Zugriffsdatum: 02.04.2013
- [18] I-NEX GMBH: *i-nex GmbH : Clevere IT-Lösungen*. <http://www.i-nex.de/home/>, . – Zugriffsdatum: 19.05.2013
- [19] IBM CORPORATION: *IBM - Rational ClearQuest*. <http://www-03.ibm.com/software/products/de/de/clearquest/>, . – Zugriffsdatum: 19.06.2013
- [20] IBM CORPORATION: *IBM - Rational Insight - Germany*. <http://www-03.ibm.com/software/products/de/de/rtl-insight/>, . – Zugriffsdatum: 04.06.2013
- [21] IBM CORPORATION: *IBM - Rational RequisitePro*. <http://www-03.ibm.com/software/products/de/de/reqpro/>, . – Zugriffsdatum: 19.06.2013
- [22] IBM CORPORATION: *IBM Domino (bekannt als Lotus Notes)*. <http://www-03.ibm.com/software/products/de/de/ibmdomino/>, . – Zugriffsdatum: 19.06.2013
- [23] IBM CORPORATION: *IBM Rational Unified Process (RUP)*. <http://www-01.ibm.com/software/rational/rup/>, . – Zugriffsdatum: 18.05.2013
- [24] KENT BECK ; MIKE BEEDLE ; ARIE VAN BENNEKUM ; ALISTAIR COCKBURN ; WARD CUNNINGHAM ; MARTIN FOWLER ; JAMES GRENNING ; JIM HIGHSMITH ; ANDREW HUNT ; RON JEFFRIES ; JON KERN ; BRIAN MARICK ; ROBERT C. MARTIN ; STEVE MELLOR ; KEN SCHWABER ; JEFF SUTHERLAND ; DAVE THOMAS: *Manifest für Agile Softwareentwicklung*. <http://agilemanifesto.org/iso/de/manifesto.html>, . – Zugriffsdatum: 19.05.2013
- [25] KENT BECK ; MIKE BEEDLE ; ARIE VAN BENNEKUM ; ALISTAIR COCKBURN ; WARD CUNNINGHAM ; MARTIN FOWLER ; JAMES GRENNING ; JIM HIGHSMITH ; ANDREW HUNT ; RON JEFFRIES ; JON KERN ; BRIAN MARICK ; ROBERT C. MARTIN ; STEVE MELLOR ; KEN SCHWABER ; JEFF SUTHERLAND ; DAVE THOMAS: *Prinzipien hinter dem Agilen Manifest*. <http://agilemanifesto.org/iso/de/principles.html>, . – Zugriffsdatum: 19.05.2013
- [26] LEHRSTUHL FÜR SOFTWARE ENGINEERING, RWTH AACHEN: *MontiArc*. <http://lab.se-rwth.de/projects/montiarc/>, . – Zugriffsdatum: 11.06.2013

- [27] LEHRSTUHL FÜR SOFTWARE ENGINEERING, RWTH AACHEN: *MontiCore*. <http://www.monticore.org/>, . – Zugriffsdatum: 11.06.2013
- [28] MACNEILL, Conor ; BODEWIG, Stefan: *Apache Ant*. <http://ant.apache.org/>, . – Zugriffsdatum 11.06.2013
- [29] MICROSOFT: *Microsoft Excel: Tabellenkalkulationssoftware*. <http://office.microsoft.com/de-de/excel/>, . – Zugriffsdatum 19.06.2013
- [30] MICROSOFT: *Microsoft Project-Produkte: Projektportfolio-Verwaltungssoftware*. <http://office.microsoft.com/de-de/project/>, . – Zugriffsdatum: 31.05.2013
- [31] PLANISWARE: *Planisware*. <http://de.planisware.com/planisware-uberblick/>, . – Zugriffsdatum: 19.06.2013
- [32] PORTER, Brett ; ZYL, Jason van: *Apache Maven*. <http://maven.apache.org/>, . – Zugriffsdatum: 11.06.2013
- [33] SAP DEUTSCHLAND: *SAP Solution Manager*. <http://www.sap.com/germany/plattform/netweaver/components/solutionmanager/index.epx>, . – Zugriffsdatum: 19.06.2013
- [34] SPOL AG, Projektmanagement: *Earned Value*. http://spol.ch/web/documents/Files/Earned_Value.pdf, . – Zugriffsdatum: 14.06.2013
- [35] TRICENTIS TECHNOLOGY & CONSULTING GMBH: *TOSCA Testsuite - Automated Software Testing*. <http://www.tricentis.com/de/solutions/>, . – Zugriffsdatum 19.06.2013
- [36] WIKIPEDIA: *Earned Value Analysis*. <http://de.wikipedia.org/w/index.php?oldid=117045372>, . – Zugriffsdatum: 14.06.2013
- [37] WIKIPEDIA: *Metrik*. <http://de.wikipedia.org/w/index.php?title=Metrik&oldid=118662475>, . – Zugriffsdatum: 03.06.2013
- [38] WIKIPEDIA: *Mind-Map*. <http://de.wikipedia.org/w/index.php?oldid=117570175>, . – Zugriffsdatum 20.06.2013

Glossar

Baseline festgelegte Konfiguration bzw. Spezifikation, welche durch die Stakeholder als Arbeitsbasis abgestimmt wurde und die nur noch mittels Change Request geändert werden kann

Change Request Anfrage einer Änderung der Baseline

Dashboard-Widget Visuelle Darstellung von Kennzahlen und anderen Informationen auf dem Dashboard

GQM-Methode Methode zur Ableitung von Metriken aus Unternehmenzielen in 2 Schritten: 1. Ableitung von Fragen bzgl. des Informationsbedürfnisses; 2. Ableitung von Metriken aus diesen Fragen. Hierbei steht GQM für **Goal-Question-Metric**

Lenkungsausschuss Oberstes Gremium bezüglich Entscheidungen innerhalb eines Projekts (beispielsweise der Ernennung des Projektleiters, etc.)

MeDIC 1. frühere Bezeichnung des *EMI*-Systems; 2. Bezeichnung der GDIS-Metrik-Datenbank

Messkunde Kunde, der die Integration eines Messsystems in Auftrag gibt

Non-Compliance Prozessabweichungen

Product Backlog Artefakt, dass alle Anforderungen eines Produktes im Scrum-Prozess beinhaltet.

SCREEN Dashboard-Komponente des Kennzahlen-Messsystems *EMI*

Scrum Prozess-Framework der agilen Entwicklung. Hierbei wird ein Produkt in kurzen Iterationszyklen (Sprint) eigenständig vom Entwicklerteam erarbeitet. Dabei entscheidet das Team eigenständig welche Arbeit getätigt wird, ohne Vorgaben eines Projektleiters zu erhalten.

Sprint Ein Iterationszyklus (meist von 2-4 Wochen) innerhalb eines Scrum-Prozesses.

Staffing Planung und Buchung von Mitarbeiter in einem Projekt.

Stakeholder Eine Person, die eine Rolle innerhalb des Projekts einnimmt (beispielsweise der Messkunde, Systemintegrator oder Projektleiter)

Use Case Anwendungsfall aus Sicht eines Akteurs

Abkürzungsverzeichnis

AC Actual Cost	GDIS Generali Deutschland Informatik Services GmbH
AGSM Advanced Generali System Engineering Method	GQM Goal-Question-Metric
BI Business Intelligence	ITIL Information Technology Infrastructure Library
CASE Computer-Aided Software Engineering	LA Lenkungsausschuss
CI Continuous Integration	LCOM Lack of Cohesion in Methods
CMMI Capability Maturity Model Integration	LOC Lines of Code
CPI Cost Performance Index	PT Personentage
CR Change Request	PV Planned Value
CV Cost Variance	QS Qualitätssicherung
EMDB Enterprise Measurement Data Bus	RUP Rational Unified Process
EMI Enterprise Measurement Infrastructure	SE System-Entwicklung
EPG Engineering Process Group	SPI Schedule Performance Index
EV Earned Value	SV Schedule Variance
FP Function Point	UML Unified Model Language

