

---

## **C\_12019\_Anlage**

---

## Inhaltsverzeichnis

<b>1 Änderungsbeschreibung .....</b>	<b>3</b>
<b>2 Änderung in gemKPT_Test.....</b>	<b>3</b>
<b>2.1 Testumgebung.....</b>	<b>4</b>
2.1.1 Vorintegrationsumgebung (RU-DEV) .....	4
2.1.2 Testumgebung (TU) .....	4
2.1.3 Referenzumgebung (RU) .....	5
<b>3 Testregelungen und Festlegungen bei Beauftragungen .....</b>	<b>6</b>
<b>4 Proof of Patient Presence PoPP .....</b>	<b>9</b>
<b>4.1 Allgemeine Anforderungen an den Test der PoPP-Service-Komponenten..</b>	<b>9</b>
4.1.1 Testumgebungen für PoPP.....	10
4.1.2 Bereitstellung PoPP-Service für die Testphase "Eigenverantwortlicher Test" ..	11
<b>4.2 Anforderungen an die Testinfrastruktur und Testartefakte der PoPP-Service-Komponenten .....</b>	<b>12</b>
<b>4.3 Anforderungen an die Testtreiberschnittstellen der PoPP-Service-Komponenten .....</b>	<b>14</b>
<b>1 Änderungsbeschreibung .....</b>	<b>3</b>
<b>2 Änderungen in gemKPT_Test .....</b>	<b>4</b>
<b>2.1 Testregelungen und Festlegungen bei Beauftragungen .....</b>	<b>6</b>
<b>2.2 Proof of Patient Presence PoPP .....</b>	<b>9</b>
2.2.1 Allgemeine Anforderungen an den Test der PoPP-Service-Komponenten.....	9
2.2.1.1 Testumgebungen für PoPP .....	10
2.2.1.2 Bereitstellung PoPP-Service für die Testphase "Eigenverantwortlicher Test" .....	11
2.2.2 Anforderungen an die Testinfrastruktur und Testartefakte der PoPP-Service-Komponenten .....	12

---

## 1 Änderungsbeschreibung

---

Die Testanforderungen für die neuen Produkttypen **PoPP-Service** und **PoPP-Modul** sind in diesem

Dieser Änderungseintrag dokumentiert. Diese Testanforderungen richten sich speziell an den Hersteller der PoPP-Komponenten bzw. PoPP-Service.

Nach Freigabe der PoPP-spezifischen Dokumente werden die Änderungen in zur [gemKPT\_Test überführt], welche die Testanforderungen für den neuen Produkttyp PoPP-Service enthalten. Produktübergreifende Anforderungen sind im Anhang C\_12020 zusammengefasst.

Änderung

---

## 2 Änderungen in gemKPT\_Test

---

**Hinweis dieses Kapitel erweitert Kapitel 3.2.10 Umgebungen in gemKPT\_Test**

### 2.1 Testumgebung

Die erfolgreiche Durchführung der definierten Testaktivitäten erfordert eine integrierte Testinfrastruktur. Alle Beteiligten im Test müssen ihren Teil der Telematikinfrastruktur als Testsystem in diese gemeinsame Umgebung einbringen. Dies gewährleistet eine realitätsnahe Simulation der Gesamtinfrastruktur und ermöglicht umfassende Tests aller Komponenten und ihrer Interaktionen.

Für die Durchführung der Tests stehen mehrere spezialisierte Umgebungen zur Verfügung:

#### 2.1.1 Vorintegrationsumgebung (RU-DEV)

Die RU-DEV ist eine flexible Entwicklungsumgebung, wo Hersteller schnell Änderungen vornehmen und testen, ohne einen formalen Änderungsprozess durchlaufen zu müssen; sie dient der Vorintegration und ermöglicht entwicklungsbegleitende Interoperabilitätstests mit Primärsystemen im Vorfeld der offiziellen Zulassung. Diese Vorstufe zur Testumgebung (TU) erfüllt folgende zentrale Funktionen:

- Frühzeitige Erkennung von Integrationsproblemen in der Telematikinfrastruktur (TI),
- Ermöglichung einer schrittweisen und gezielten Integration von Primärsystemen und Anwendungen,
- Identifikation potenzieller Interoperabilitätsprobleme vor Beginn der offiziellen Zulassungsphase,
- Bereitstellung einer Testplattform für Primärsystemhersteller zur iterativen Produktintegration,
- Schaffung eines kontrollierten Raums für Tests und notwendige Anpassungen vor der Überführung in die TU.

Für diese Testumgebungen können auch Simulatoren und Mocks eingesetzt werden, falls Echtkomponenten noch nicht bereitgestellt werden können.

#### 2.1.2 Testumgebung (TU)

Die Testumgebung (TU) ist eine zentrale Komponente in der Testinfrastruktur der gematik, speziell ausgelegt für die Durchführung umfassender Ende-zu-Ende (E2E) und Interoperabilitätstests. Sie dient als realitätsnahe Simulation der gesamten Telematikinfrastruktur und ermöglicht die Integration und Prüfung aller relevanten Produkte in einem kontrollierten Umfeld.

Die TU wird primär für die Durchführung von funktionalen Zulassungstests genutzt.

Um die Integrität der Umgebung sicherzustellen sind strenge und produktionsnahe Zugangskontrollen und Sicherheitsmaßnahmen eingeführt.

### **2.1.3 Referenzumgebung (RU)**

Diese Umgebung dient als Referenz für die Telematikinfrastruktur und unterstützt folgende Szenarien:

Durchführung spezialisierter nicht-funktionaler Tests

- Reproduktion von Fehlern aus der Produktivumgebung
- Ausführung von HotFix und Emergency Tests
- Bereitstellung von Serviceprodukten durch die gematik
- Unterstützung bei der Entwicklung neuer Anwendungen für Dritte
- Förderung der "Ende-zu-Ende-Verantwortung" von Herstellern und Anbietern

Die RU wird von Herstellern und interessierten Parteien, wie Universitäten, für Schulungszwecke genutzt. Der dezentrale Bereich wird von Industriepartnern (Enabler) unabhängig organisiert. Dies ermöglicht eine flexible und praxisnahe Nutzung der Umgebung für verschiedene Zwecke innerhalb des TI.

#### **A\_27249—Referenzobjekte in der Test und Referenzumgebungen**

Der Hersteller eines zugelassenen Produkts der Telematikinfrastruktur (TI) ist verpflichtet, dieses Produkt als Referenzobjekt in der Testumgebung TU und Referenzumgebung RU bereitzustellen und zu pflegen.

Die Bereitstellung und Pflege des Referenzobjekts beginnt mit der Zulassungserteilung des Produkts und endet mit der offiziellen Außerbetriebnahme in der Produktivumgebung (PU) der Telematikinfrastruktur.

Der Hersteller muss das Referenzobjekt stets auf dem aktuellsten Stand halten, der dem in der PU eingesetzten Produktstand entspricht.

[<=]

#### **A\_27250—Testobjekte in der Test und Referenzumgebungen**

Der Hersteller eines für die Telematikinfrastruktur (TI) entwickelten Produkt ist verpflichtet, dieses Produkt als Testobjekt in der Testumgebung TU und Referenzumgebung RU bereitzustellen und zu pflegen ab Beginn der Zulassungstestphase.

Die Bereitstellung und Pflege des Referenzobjekts beginnt mit der Start der Zulassungstestphase und endet mit der offiziellen Zulassung oder der Entscheidung, das Produkt nicht weiter für eine Zulassung zu verfolgen.

[<=]

#### **A\_27262—Vorgaben an die Deployments in der RU-DEV-Umgebung**

Deployments in der RU-DEV finden nur in Abstimmung mit dem Test-Consultant der gematik statt und MÜSSEN mit folgenden Informationen begleitet werden:

- Versionsnummer oder Build-Identifizier: Zur eindeutigen Identifikation des Deployment,
- Änderungsprotokoll (Changelog): Eine Zusammenfassung der implementierten Funktionen und Fehlerbehebungen,
- Konfigurationsdetails: Informationen zu Umgebungsvariablen und spezifischen Einstellungen für die Entwicklungsumgebung,

- ~~Abhängigkeiten: Liste aller externen Dienste, Bibliotheken oder APIs, die für das Deployment erforderlich sind,~~
- ~~Bereitstellungszeitpunkt: Datum und Uhrzeit des Deployments,~~
- ~~Rollback-Anweisungen: Für die Wiederherstellung des vorherigen Zustands im Falle von Probleme,~~
- ~~Bekannte Probleme oder Einschränkungen: Dokumentation eventueller bekannter Fehler oder Limitierungen.~~

~~{<=>}~~

## 2.22.1 Testregelungen und Festlegungen bei Beauftragungen

Dieses Kapitel wird vor Kapitel 6.1 in gemKPT\_Test übernommen und enthält spezielle Anforderungen für Testregelungen und Festlegungen bei Beauftragungen.

Die Test- und Prüfkativitäten der gematik bei Beauftragungen zielen auf eine umfassende Validierung der Qualität, Funktionalität und Interoperabilität der entwickelten Produkte und Anwendungen im Gesamtkontext der Telematikinfrastruktur ab.

Dieser ganzheitliche Ansatz gewährleistet, dass die Produkte sowohl eigenständig funktionieren als auch nahtlos mit anderen Komponenten interagieren und die vorgegebenen Qualitätsstandards erfüllen. Der Testprozess umfasst produktspezifische und -übergreifende Tests, um die Ende-zu-Ende-Funktionalität und Interoperabilität im gesamten System sicherzustellen. Der Auftragnehmer (AN) muss die Erfüllung aller funktionalen und nicht-funktionalen Anforderungen nachweisen.

Während der Entwicklung ist der AN verpflichtet, den Fortschritt an definierten Kontrollpunkten transparent darzulegen und das Erreichen von Meilensteinen proaktiv der gematik mitzuteilen. Nach Abschluss der Entwicklung erfolgt eine Güteprüfung durch die gematik, bei der die vom AN erstellte Testdokumentation eingehend bewertet wird. In der Gesamtintegration testet die gematik das Zusammenspiel aller Komponenten der TI.

Die erfolgreiche Gesamtintegration ist Voraussetzung für die Produktivsetzung, eine formale Abnahme wird im jeweiligen Vertrag mit dem AN geregelt.

Der Test Consultant begleitet den gesamten Prozess von der Beauftragung bis zur Abnahme und übernimmt verschiedene Verantwortlichkeiten. Folgende Tätigkeiten sind relevant:

1. Kontrollpunkte
  - Ziel: Überprüfung des Projektfortschritts, regelmäßige Bewertung von Zwischenergebnissen, Möglichkeit für zeitnahe Korrekturen
  - Art: Informativ
  - Aufgabe Test Consultant: Begleitend
2. Information
  - Ziel: Kontinuierlicher Austausch mit dem Auftragnehmer, proaktive Mitteilung über Meilensteine und potenzielle Blocker vom Auftragnehmer, Sicherstellung der Transparenz im Projektverlauf
  - Art: Informativ
  - Aufgabe Test Consultant: Begleitend

3. Güteprüfung
  - Ziel: Detaillierte Bewertung der Testdokumentation
  - Art: Formal
  - Aufgabe Test Consultant: Prüfung
4. Zulassungstests
  - Ziel: Gesamtintegration in der Telematikinfrastruktur
  - Art: Test
  - Aufgabe Test Consultant: Durchführend
5. Abnahme
  - Ziel: Überprüfung und Freigabe der Gesamtleistung
  - Art: Formal
  - Aufgabe Test Consultant: Begleitend

#### ~~A\_27248—Güteprüfung Test~~

#### A\_27248 -Güteprüfung Test

Die Güteprüfung stellt eine zentrale Qualitätssicherungsmaßnahme im Rahmen von Beauftragungen dar. Sie dient der detaillierten Prüfung und Bewertung der geforderten Testdokumente und die jeweiligen Produkte.

Der Auftragnehmer MUSS mindestens folgende Testdokumente als Liefergegenstände bereitzustellen:

1. Testkonzept: Entsprechend den Vorgaben in Tab\_Test\_013 des [gemKPT\_Test],
2. Testspezifikation: Gemäß den Anforderungen in Tab\_Test\_014 des [gemKPT\_Test],
3. Testbericht: Konform zu Tab\_Test\_018 des [gemKPT\_Test],
4. Produktdokumentation: entsprechend der Vorgaben in "Tab\_Test\_016 Produktdokumentation" des [gemKPT\_Test],
5. Afo-Testmatrix: In Übereinstimmung mit den Richtlinien des [gemKPT\_Test].

Diese Dokumente bilden die Grundlage für eine umfassende Beurteilung der Testqualität und -Vollständigkeit. Der Auftraggeber (AG) prüft die Dokumente auf Einhaltung der definierten Vorgaben gemäß [TIP1-A6524-01] und protokolliert die Prüfung.

[<=]

-{<=}

#### ~~A\_27141—Testmanagementsystem des AN~~

#### A\_27141 -Testmanagementsystem des AN

Der AN MUSS für die Testaktivitäten während der eigenverantwortlichen Tests ein umfassendes Testmanagementsystem einsetzen und dem AG den vollen Zugriff hierfür gewähren. Das Testmanagementsystem MUSS folgende Testaktivitäten unterstützen:

1. **Testdurchführung:** Alle Tests im System durchführen und dokumentieren, einschließlich E2E-Tests und BDD-Szenarien.
2. **Fehlermanagement:** Fehler erfassen, mit Details versehen und sichtbar machen.
3. **Fortschritt und Berichte:** Testfortschritt jederzeit einsehbar, automatische Berichtgenerierung.

4. **Artefakte:** Testartefakte im System verwalten.
5. **Dokumentation:** Benutzerdokumentation und Systemanleitungen bereitstellen.
6. **Zugriffsrechte:** AG erhält Lesezugriff, bei Bedarf auch Schreibrechte.
7. **Versionskontrolle:** Für Testfälle, Skripte und andere Artefakte.
8. **Nachverfolgbarkeit:** Zwischen Anforderungen, Testfällen und Ergebnissen.

Der AN MUSS die Erfüllung dieser Anforderungen für eine transparente Zusammenarbeit gewährleisten.

[<=]

{<=}

#### ~~A\_27142—Anforderungen an Testreporting~~

#### ~~A\_27142 -Anforderungen an Testreporting~~

Der AN MUSS über den Teststatus regelmäßig berichten (Frequenz noch zu definieren), dabei wird:

1. Eine Darstellung des aktuellen Testfortschritts, einschließlich der Anzahl durchgeführter, erfolgreicher und fehlgeschlagener Tests.
2. Einer detaillierten Erfassung aller gefundenen Fehler, inklusive Schweregrad, Priorität und Status.
3. Informationen zur Testabdeckung, um sicherzustellen, dass alle kritischen Funktionen getestet wurden.
4. Eine Einschätzung der identifizierten Risiken basierend auf den Testergebnissen.

[<=]

#### ~~A\_27142-01 -Anforderungen an Testreporting~~

Der AN MUSS über den Teststatus der eigenverantwortlichen Tests in der RU DEV regelmäßig berichten (Frequenz noch zu definieren), dabei werden folgende Informationen bereitgestellt:

1. Eine Darstellung des aktuellen Testfortschritts, einschließlich der Anzahl durchgeführter, erfolgreicher und fehlgeschlagener Tests.
2. Einer detaillierten Erfassung aller gefundenen Fehler, inklusive Schweregrad, Priorität und Status.
3. Informationen zur Testabdeckung, um sicherzustellen, dass alle kritischen Funktionen getestet wurden.
4. Eine Einschätzung der identifizierten Risiken basierend auf den Testergebnissen.

[<=]

{<=}

~~Dieses Kapitel wird nach Kapitel 9 in gemKPT\_Test übernommen, und beinhaltet spezielle Anforderungen für den PoPP-Service~~



## ~~2.3.2~~ 2.3.2 Proof of Patient Presence PoPP

Dieses Kapitel wird nach Kapitel 9 in gemKPT\_Test übernommen und enthält spezielle Anforderungen für den PoPP -Service.

Die Telematikinfrastruktur erfährt im Zuge ihrer Modernisierung auf Version 2.0 diverse architektonische Verbesserungen.

Ein Kernaspekt der TI 2.0 ist die Nutzung der TI-Fachdienste "Anytime Anywhere" über das Internet, ~~wonach ein Versorgungskontext zwischen Versichertem und Leistungserbringer nachgewiesen wird. Der dafür zuständige.~~

Der PoPP-Service erzeugt die Bestätigung eines Versorgungskontextes in Form des kryptographisch gesicherten PoPP-Token. Dieses bestätigt, dass ein bestimmter Versicherter mit einer bestimmten Leistungserbringerinstitution zusammengekommen ist. ~~Dieser Versorgungskontext~~ Der PoPP-Token wird beim Zugriff von Leistungserbringern auf TI-Fachdienste (z. B. VSDM2, ePA für alle, E-Rezept) ~~diesen~~ mit übergeben.

### ~~2.3.1~~ 2.2.1 Allgemeine Anforderungen an den Test der PoPP-Service-Komponenten

#### ~~A\_26671—Erfüllung der Anforderungen aus dem Produkttypsteckbrief~~

#### ~~A\_26673 -Testumgebung in der Produktentwicklungsphase~~

Der Hersteller des PoPP-Services MUSS alle funktionalen Anforderungen aus dem Produkttypsteckbrief ~~[gemProdT\_PoPP\_Service\_PTV]~~ durch Tests nachweisen. Die Tests von funktionalen Anforderungen, die nicht über die externen Produktschnittstellen getestet werden können, können mittels Testtreiberschnittstellen durchgeführt werden. ~~[<=]~~

#### ~~A\_26673—Testumgebung in der Produktentwicklungsphase~~

Der Hersteller des PoPP-Services MUSS in der Produktentwicklungsphase ~~die PoPP-~~ ~~Produkt~~ das Testobjekt in der Vorintegrationsumgebung RU DEV bereitstellen, um Tests für die Kontrollpunkte unter [A\_26676] zu ermöglichen.

~~[<=]~~

-

~~[<=]~~

#### ~~A\_26674—Bereitstellung von Testkomponenten und Testartefakten in den Testumgebungen~~

#### ~~A\_26674 -Bereitstellung von Testkomponenten und Testartefakten in den Testumgebungen~~

Der Hersteller des PoPP-Services MUSS ~~Testkomponente~~ das Testobjekt und Testartefakte so entwickeln und bereitstellen, dass diese in allen Testumgebungen genutzt werden können.

~~[<=]~~

~~[<=]~~

#### ~~A\_26675—Format der Testszenarien~~

#### ~~A\_26675 -Format der Testszenarien~~

Der Hersteller des PoPP-Services MUSS alle für den funktionalen Test erforderlichen Testszenarien und Testfälle nach der Behavior Driven Development (BDD)-

Technik erstellen und z. B. in der Beschreibungssprache "Gherkin" formulieren.

[<=]

{<=}

#### ~~A\_26676—Kontrollpunkte während der Entwicklungsphase~~

#### **A\_26676 -Kontrollpunkte während der Entwicklungsphase**

Der AN MUSS die ~~vom AG~~ von der gematik vorgegebenen Kontrollpunkte während der Entwicklungsphase durchführen. Dadurch wird die Reife und der Fortschritt der in Entwicklung befindlichen Produktkomponenten überprüft.

[<=]

{<=}

#### ~~A\_26677—Herstellerspezifische Kontrollpunkte~~

#### **A\_26677 -Herstellerspezifische Kontrollpunkte**

Der Hersteller des PoPP-~~Services~~-Service KANN weitere Kontrollpunkte in seinem Qualitätsprozess vorsehen und umsetzen. In diesem Fall KÖNNEN die Testergebnisse in einer von der gematik bereitgestellten Datenaustauschplattform abgelegt werden.

[<=]

{<=}

#### ~~A\_26678—Tests der einzelnen Systemkomponenten über die externe Schnittstelle~~

Der AN MUSS ~~für die Komponententests alle darin verorteten funktionalen Anforderungen über die dokumentierte externe Schnittstelle testen.~~

{<=}

#### ~~A\_26679—Tests der einzelnen Systemkomponenten über Testtreiber~~

Der

#### **A\_26680 -Versionierung der Testkomponenten**

Der Hersteller des PoPP-~~Services~~ MUSS ~~eine Testtreiberschnittstelle entwickeln und für die Tests der funktionalen Anforderungen einsetzen, die in den Systemkomponenten nicht über ihren externen Schnittstellen testen kann.~~

{<=}

#### ~~A\_26680—Versionierung der Testkomponenten~~

Der Hersteller des PoPP-~~Services~~-Service MUSS alle für den Test genutzten Komponenten mit Datum und Version auflisten.

[<=]

-

{<=}

#### **A\_26680-01 -Versionierung der Testkomponenten**

Der AN MUSS alle für den Test genutzten Komponenten mit Datum und Version auflisten.

[<=]

### ~~2.3.1.1~~ 2.2.1.1 Testumgebungen für PoPP

Die neue Teststrategie der gematik unterscheidet zwischen einer Phase "Eigenverantwortlicher Test (EvT)", welcher durch den Hersteller des PoPP-Service selbst und einer Phase "Zulassungstest (ZuT)", welcher durch die gematik im Zusammenspiel

mit dem Hersteller des PoPP-Service durchgeführt wird. Diese Testphasen werden in verschiedenen Testumgebungen absolviert, für die der Hersteller des PoPP-Service Testobjekte (Produktmuster, Zulassungsobjekt) zur Verfügung stellen müssen. Die folgenden Systemumgebungen werden für den Test des PoPP-Service mindestens benötigt:

**Tabelle 1: Testumgebungen für PoPP**

Bezeichnung	Verantwortung	Testphase	Teststufe(n)	Testobjekt
DEV	Hersteller	EvT	Produkttest	Hersteller
RU DEV	gematik	EvT	Produktübergreifender TestInteroperabilitätstest (IOP)	Hersteller
TU	gematik	ZuIT	Produktübergreifender Test, Gesamtintegrationstest	Betreiber
RU	gematik	ZuIT	Gesamtintegrationstest, Hotfix-Test	Betreiber

### **2.3.1.22.2.1.2 Bereitstellung PoPP-Service für die Testphase "Eigenverantwortlicher Test"**

Der Hersteller des PoPP-Service soll zukünftig die Möglichkeit erhalten, produktübergreifende, eigenverantwortliche Tests (EvT-Püt) IOP noch vor den Zulassungstests ausführen zu können. Hierzu stellt die gematik eine neue Testumgebung bereit, die RU DEV. Für die Bereitstellung des PoPP-Service in der RU DEV während der Entwicklungsphase ist der Hersteller des PoPP-Service verantwortlich. In der RU DEV können E2E-Testszzenarien für PoPP und Interoperabilität auch mit Primärsystemherstellern getestet werden.

Die gematik koordiniert die Entwicklung einer Testsuite, welche auf gitHub zur gemeinsamen Weiterentwicklung und Nutzung zur Verfügung gestellt wird. Die Testsuite bietet die Möglichkeit, die wichtigsten E2E-Testszzenarien für PoPP automatisch ausführbar zu machen. Der erfolgreiche Testdurchlauf gilt als Vorbedingung für den Einsatz des Produktes in der jeweiligen Testumgebung.

#### **A\_27393—Bereitstellung als signiertes Container-Image**

#### **A\_27393 -Bereitstellung als signiertes OCI konformes Image**

Der Hersteller des PoPP-ServiceService MUSS jedes Release des PoPP-ServiceTestobjektes als signiertes Container-OCI konformes Image gemeinsam mit der Konfigurationsanleitung zur Integration in die gemeinsame EvT-Püt-RU DEV Umgebung liefern. [ <= ]

·[ <= ]

#### **A\_27394—Agiler Bereitstellungsprozess**

#### **A\_27394 -Agiler Bereitstellungsprozess**

Um die Stabilität der gemeinsamen EvT-Püt-UmgebungenRU DEV Umgebung zu gewährleisten, MUSS der Hersteller des PoPP-ServiceService sich an folgendem agilen Prozess halten.

1. ~~Agiler Bereitstellungsprozess~~—Änderungsmanagement: Der ~~Hersteller des~~ PoPP-Service-~~Hersteller~~ DARF Modifikationen an den Komponenten ausschließlich innerhalb der dafür vorgesehenen Zeitfenster zwischen den durch gematik festgelegten Sprints vornehmen.
2. ~~Agiler Bereitstellungsprozess~~—Flexibilität bei der Bereitstellung: Der ~~Hersteller des~~ PoPP-Service ~~Hersteller~~ MUSS das aktuelle Bereitstellungsfenster überspringen, wenn ~~der PoPP-Service~~ das ~~Testobjekt~~ noch keinen stabilen Status erreicht hat.
3. ~~Agiler Bereitstellungsprozess~~—Automatisierte Qualitätssicherung: Nach dem Schließen des Änderungszeitfensters werden in der ~~EVT-PÜT-RU DEV~~ Umgebung über Nacht automatisierte E2E-Tests ausgeführt. Diese haben zum Ziel, potenzielle Probleme in den Komponenten zu identifizieren, die möglicherweise negative Auswirkungen auf die Tests anderer Hersteller haben könnten.
4. ~~Agiler Bereitstellungsprozess~~—Fehleranalyse-Phase: Der ~~Hersteller des~~ PoPP-Service-~~Hersteller~~ MUSS während der Testdurchführung aufgetretene Fehler, nach der Bereitstellung analysieren, um die betroffenen Komponenten zu identifizieren. Die gematik legt die Länge der Fehleranalyse-Phase anlassbezogen fest.
5. ~~Agiler Bereitstellungsprozess~~—Rollback: Der ~~Hersteller des~~ PoPP-Service ~~Hersteller~~ MUSS ~~den PoPP-Service~~ das ~~Testobjekt~~ am Ende der Fehleranalyse-Phase auf die letzte bekannte, fehlerfreie Version zurückzusetzen, falls Fehler im ~~PoPP-Service~~ ~~Testobjekt~~ festgestellt wurden, welche die Integrität des Gesamtsystems gefährden. Der ~~PoPP-Service~~ Hersteller ~~des PoPP-Service~~ MUSS die festgestellten Fehler im ~~PoPP-Service~~ ~~Testobjekt~~ in den nächsten Sprints beheben.
6. ~~Agiler Bereitstellungsprozess~~—Abschluss der Änderungen: Nach dem Rollback prüft die automatisierte Qualitätssicherung erneut, ob die Stabilität der gemeinsamen Umgebung gewährleistet ist. Sollte dies nicht der Fall sein, werden die Schritte 4 bis 6 solange überprüft, bis die gemeinsame Umgebung die notwendige Stabilität erreicht.

Durch diese strukturierte Vorgehensweise wird eine hohe Systemstabilität gewährleistet und gleichzeitig die notwendige Flexibilität für kontinuierliche Verbesserungen und Anpassungen ermöglicht.

[<=]

{<=}

## 2.3.22.2.2 Anforderungen an die Testinfrastruktur und Testartefakte der PoPP-Service-Komponenten

### ~~A\_26684—Freie Nutzung der entwickelten Testartefakte~~

#### ~~A\_26684 -Freie Nutzung der entwickelten Testartefakte~~

Der Hersteller des PoPP-~~Services~~Service MUSS alle Testartefakte, die für die Testaktivitäten vom Hersteller entwickelt werden, der gematik Kosten- und Lizenzfrei zur Nutzung und Weiterentwicklung zur Verfügung stellen.

[<=]

{<=}

### ~~A\_27053—Freie Nutzung der erworbenen Testartefakte~~

**A\_27053 -Freie Nutzung der erworbenen Testartefakte**

Der Hersteller des PoPP-~~Services~~Service MUSS alle Testartefakte, die für die Testaktivitäten von Dritten erworben werden, der gematik Kosten- und lizenzfrei zur Nutzung zur Verfügung stellen.

[<=]

{<=}

**~~A\_26685—Automatisierung der Testartefakte~~****A\_26685 -Automatisierung der Testartefakte**

Der Hersteller des PoPP-~~Services~~Service MUSS alle Testartefakte so entwickeln und bereitstellen, dass diese eine kontinuierliche, automatisierte Testausführung ermöglichen.

[<=]

{<=}

**~~A\_26686—Verwendbarkeit von Testkomponenten und Testartefakten in automatisierten CI/CD-Pipelines~~****A\_26686 -Verwendbarkeit von Testkomponenten und Testartefakten in automatisierten CI/CD-Pipelines**

Der Hersteller des PoPP-~~Services~~Service MUSS ausführbare Testartefakte so entwickeln und bereitstellen, dass sie in automatisierten CI/CD-Pipelines verwendet werden können.

[<=]

{<=}

**~~A\_26688—Labeln von Komponenten~~****A\_26688 -Labeln von Komponenten**

Der Hersteller des PoPP-~~Services~~Service MUSS seine Komponenten mit einem zur Umgebung gehörenden Label versehen. Dadurch sollen Komponenten im Produktlebenszyklus eindeutig identifiziert werden, ob sie in der TU oder PU eingesetzt werden dürfen.

[<=]

{<=}

**~~A\_26689—Keine Hardware-Abhängigkeiten bei Komponenten in Testumgebungen~~****A\_26689 -Keine Hardware-Abhängigkeiten bei Komponenten in Testumgebungen**

Der Hersteller des PoPP-~~Services~~Service MUSS seine Komponenten so entwickeln und bereitstellen, dass keine Hardware-Abhängigkeiten, z. B. zu einem HSM entstehen. Dies kann z. B. durch den Einsatz von Simulatoren oder Mocks erreicht werden.

[<=]

{<=}

**~~A\_26690—Nutzung von Private oder Secret Keys in Testumgebungen~~****A\_26690 -Nutzung von Private oder Secret Keys in Testumgebungen**

Der Hersteller des PoPP-~~Services~~Service MUSS in Testumgebungen sicherstellen, dass die verwendeten Private oder Secret Keys bekannt sind.

[<=]

{<=}

*Hinweis: Dies kann z. B. durch das Einbringen von bekannten Private bzw. Secret Keys erfolgen, oder durch dessen Auslesen aus einem Schlüsselspeicher der Testumgebung.*

**~~A\_27100—Übergreifende Testmaßnahmen~~**

**A\_27100 -Übergreifende Testmaßnahmen**

Der Hersteller des PoPP-ServicesService MUSS bei der Vorbereitung und Durchführung von übergreifenden Testmaßnahmen, wie z. B. Durchstichtests oder Connectathons, auch nach der Zulassung auf Aufforderung des AG mitwirken, z. B. durch Bereitstellung und Konfiguration der benötigten Komponenten und Testumgebungen oder Unterstützung bei der Problemanalyse. [ <= ]

[ <= ]

*Eine genaue Definition von Art und Umfang des zu leistenden Supports wird vertraglich zwischen gematik und Hersteller abgestimmt.*

**2.4 Anforderungen an die Testtreiberschnittstellen der PoPP-Service-Komponenten**

Für die Erfüllung der Spezifikationsanforderungen kommunizieren die PoPP-Komponenten untereinander und mit externen Instanzen wie z. B. Primärsysteme oder weitere TI-Dienste. Nicht alle funktionalen Anforderungen, die in den internen Komponenten umgesetzt werden, können über eine API-Schnittstelle getestet werden; dadurch kann eine vollständige Testabdeckung nicht garantiert werden. Für die Sicherstellung der Produktqualität ist es notwendig, in den entsprechenden PoPP-Komponenten eine Testtreiber-Schnittstelle bzw. einen Testmodus zu implementieren, der eine vollständige Testabdeckung ermöglicht.

**A\_26694 – Bereitstellen einer Testtreiberschnittstelle für den Testbetrieb**

Der Hersteller des PoPP-Services MUSS in jeder Komponente eine Testtreiberschnittstelle bereitstellen, die es ermöglicht, in einem Blackbox-Test, d.h. einem Test an den externen Schnittstellen, automatisiert alle funktionalen Anforderungen an die jeweilige Komponente zu prüfen, die sonst über die normale API der Komponente nicht oder nur mit erhöhtem Aufwand prüfbar wären.

[ <= ]

**A\_26695 – Keine Testtreiberschnittstelle in produktiv einsetzbaren Komponenten**

Der Hersteller der PoPP-Services MUSS sicherstellen, dass die Testtreiberschnittstelle in produktiven Umgebungen nicht enthalten ist.

[ <= ]

**A\_26696 – API und Dokumentation der Testtreiberschnittstelle**

Der Hersteller des PoPP-Services MUSS die API der Testtreiberschnittstelle und ihre Dokumentation frei verfügbar bereitstellen.

[ <= ]

*Hinweis: Eine genaue Abstimmung zu Art und Umfang der Testtreiberschnittstelle in den jeweiligen Komponenten erfolgt später in Abstimmung mit dem Hersteller.*