Verwendung von Aufzeichnungen über architektonische Entscheidungen zur Optimierung der technischen Entscheidungsfindung für ein Softwareentwicklungsprojekt

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Verwendung von Aufzeichnungen über architektonische Entscheidungen zur Optimierung der technischen Entscheidungsfindung für ein Softwareentwicklungsprojekt

Table of Contents

Verwendung von Aufzeichnungen über architektonische Entscheidungen zur Optimierung der technischen Entscheidungsfindung für ein Softwareentwicklungsprojekt

Darius Kunce und Dominik Goby, Amazon Web Services (AWS)

März 2022 (Dokumentverlauf)

In diesem Leitfaden wird der ADR-Prozess (Architectural Decision Records) für Softwareentwicklungsprojekte vorgestellt. ADRs unterstützt die Teamausrichtung, dokumentiert strategische Leitlinien für ein Projekt oder Produkt und reduziert wiederkehrende und zeitaufwändige Entscheidungsprozesse.

Während der Projekt- und Produktentwicklung müssen Softwareentwicklungsteams architektonische Entscheidungen treffen, um ihre Ziele zu erreichen. Diese Entscheidungen können technischer Natur sein, z. B. die Entscheidung, das CQRS-Muster (Command Query Responsibility Segregation) zu verwenden, oder prozessbezogener Art sein, z. B. die Entscheidung, den GitFlow Workflow zur Verwaltung des Quellcodes zu verwenden. Diese Entscheidungen zu treffen ist ein zeitaufwändiger und schwieriger Prozess. Die Teams müssen diese Entscheidungen begründen, dokumentieren und den relevanten Stakeholdern mitteilen.

Bei architektonischen Entscheidungen treten häufig drei wichtige Anti-Muster zutage:

- Aus Angst, die falsche Wahl zu treffen, wird überhaupt keine Entscheidung getroffen.
- Eine Entscheidung wird ohne jegliche Begründung getroffen, und die Leute verstehen nicht, warum sie getroffen wurde. Dies führt dazu, dass dasselbe Thema mehrfach erörtert wird.
- Die Entscheidung wird nicht in einem Archiv für architektonische Entscheidungen gespeichert, sodass die Teammitglieder vergessen oder nicht wissen, dass die Entscheidung getroffen wurde.

Es ist besonders wichtig, diese Antimuster während des Entwicklungsprozesses eines Produkts oder Projekts zu beseitigen.

Die Erfassung der Entscheidung, des Kontextes und der Überlegungen, die zu der Entscheidung geführt haben, in Form einer alternativen Streitbeilegung ermöglicht es aktuellen und zukünftigen Stakeholdern, Informationen über die getroffenen Entscheidungen und den Denkprozess hinter

jeder Entscheidung zu sammeln. Dies reduziert die Zeit für die Softwareentwicklung und bietet eine bessere Dokumentation für zukünftige Teams.

Gezielte Geschäftsergebnisse

ADRs drei Geschäftsergebnisse anstreben:

- Sie bringen aktuelle und zukünftige Teammitglieder zusammen.
- Sie legen eine strategische Ausrichtung für das Projekt oder Produkt fest.
- Sie vermeiden Entscheidungsmuster, indem sie einen Prozess definieren, mit dem architektonische Entscheidungen ordnungsgemäß dokumentiert und kommuniziert werden.

ADRs Erfassen Sie den Kontext der Entscheidung, um future Interessengruppen zu informieren. Eine Sammlung von Erfahrungsberichten und Referenzdokumenten ADRs zur Verfügung stellen. Team- oder Projektmitglieder verwenden die ADR-Sammlung für Folgeprojekte und die Planung von Produktfeatures. Durch die Möglichkeit, auf Referenzen zurückgreifen zu können, wird der Zeitaufwand für die Entwicklung, Überprüfung und architektonische Entscheidungen ADRs reduziert. ADRs ermöglicht es auch anderen Teams, von den Überlegungen anderer Projekt- und Produktentwicklungsteams zu lernen und Einblicke in diese zu gewinnen.

ADR-Prozess

Ein Architectural Decision Record (ADR) ist ein Dokument, das eine Entscheidung beschreibt, die das Team in Bezug auf einen wichtigen Aspekt der Softwarearchitektur trifft, die es erstellen möchte. Jedes ADR beschreibt die architektonische Entscheidung, ihren Kontext und ihre Konsequenzen. ADRs haben Status und folgen daher einem Lebenszyklus. Ein Beispiel für eine Beispiel für eine solche ADR finden Sie im Anhang.

Das ADR-Verfahren gibt eine Sammlung von Aufzeichnungen über architektonische Entscheidungen aus. Diese Sammlung erstellt das Entscheidungsprotokoll. Das Entscheidungsprotokoll enthält den Projektkontext sowie detaillierte Implementierungs- und Entwurfsinformationen. Die Projektmitglieder überfliegen die Schlagzeilen der einzelnen ADR, um sich einen Überblick über den Projektkontext zu verschaffen. Sie lesen die ADRs, um sich eingehend mit Projektimplementierungen und Entwurfsentscheidungen auseinanderzusetzen.

Wenn das Team ein ADR akzeptiert, wird es unveränderlich. Wenn neue Erkenntnisse eine andere Entscheidung erfordern, schlägt das Team ein neues ADR vor. Wenn das Team das neue ADR akzeptiert, ersetzt es das vorherige ADR.

Umfang des ADR-Verfahrens

Die Projektmitglieder sollten für jede architektonisch bedeutsame Entscheidung, die sich auf das Softwareprojekt oder das Softwareprodukt auswirkt, ein ADR erstellen, einschließlich der folgenden (Richards und Ford 2020):

- Struktur (zum Beispiel Muster wie Microservices)
- Non-functional Anforderungen (Sicherheit, Hochverfügbarkeit und Fehlertoleranz)
- Abhängigkeiten (Kopplung von Komponenten)
- Schnittstellen (APIs und veröffentlichte Verträge)
- Konstruktionstechniken (Bibliotheken, Frameworks, Tools und Prozesse)

Funktionale und nichtfunktionale Anforderungen sind die häufigsten Eingaben für den ADR-Prozess.

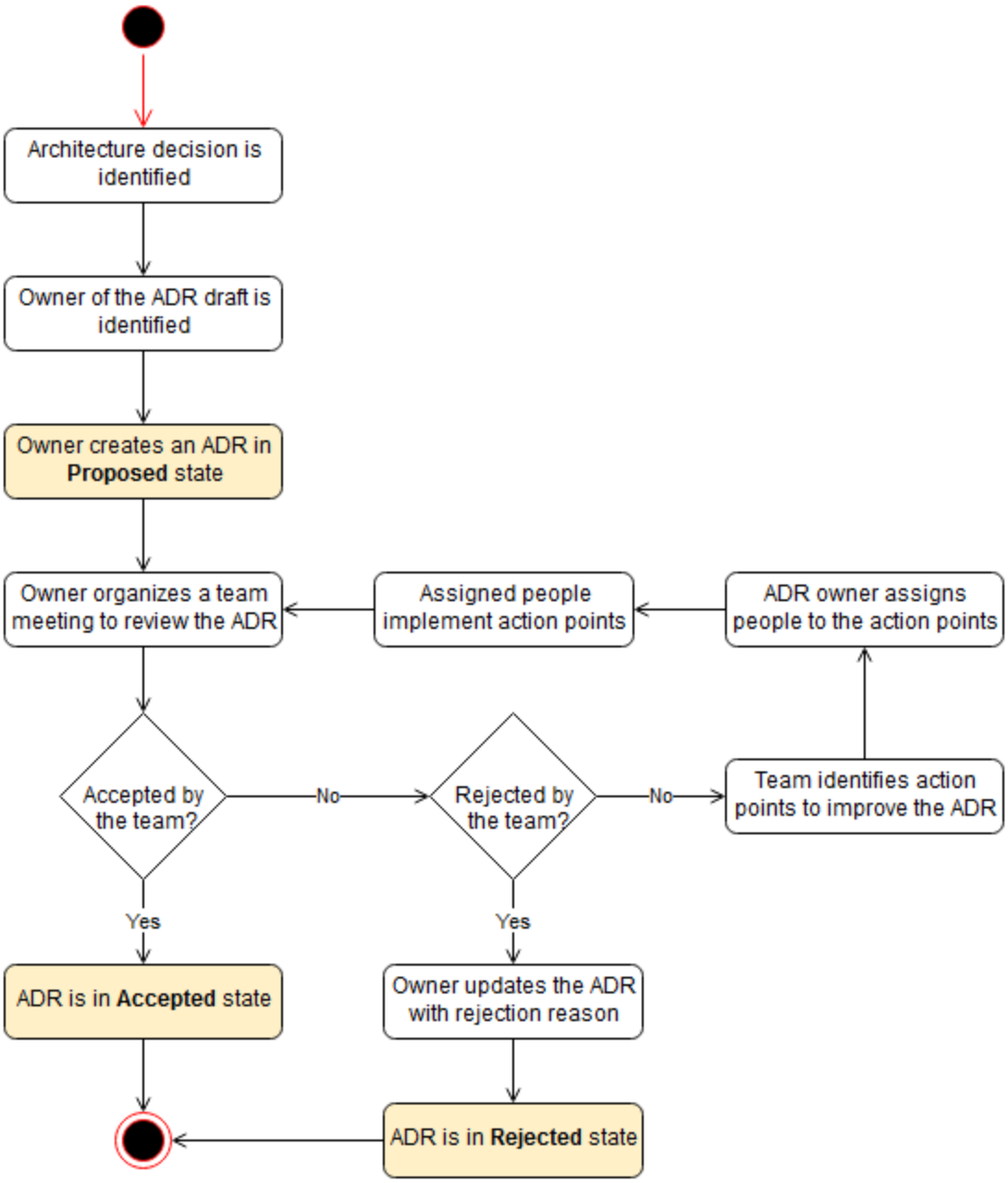
ADR-Inhalte

Wenn das Team feststellt, dass ein ADR erforderlich ist, beginnt ein Teammitglied, das ADR auf der Grundlage einer projektweiten Vorlage zu schreiben. (Beispielvorlagen finden Sie in der GitHubADR-Organisation.) Die Vorlage vereinfacht die ADR-Erstellung und stellt sicher, dass das ADR alle relevanten Informationen erfasst. Jede alternative Streitbeilegung sollte mindestens den Kontext der Entscheidung, die Entscheidung selbst und die Konsequenzen der Entscheidung für das Projekt und seine Ergebnisse definieren. (Beispiele für diese Abschnitte finden Sie im Anhang.) Einer der wichtigsten Aspekte der ADR-Struktur besteht darin, dass sie sich auf den Grund der Entscheidung konzentriert und nicht darauf, wie das Team sie umgesetzt hat. Zu verstehen, warum das Team die Entscheidung getroffen hat, erleichtert es anderen Teammitgliedern, die Entscheidung zu treffen, und verhindert, dass andere Architekten, die nicht am Entscheidungsprozess beteiligt waren, diese Entscheidung in Zukunft außer Kraft setzen.

Verfahren für die Einführung von ADR

Jedes Teammitglied kann ein ADR erstellen, aber das Team sollte eine Definition der Eigentümerschaft für ein ADR festlegen. Jeder Autor, der Eigentümer eines ADR ist, sollte den ADR-Inhalt aktiv pflegen und kommunizieren. Um diese Eigentumsverhältnisse zu verdeutlichen, werden in diesem Leitfaden die ADR-Autoren in den folgenden Abschnitten als ADR-Eigentümer bezeichnet. Andere Teammitglieder können jederzeit zu einem ADR-Verfahren beitragen. Wenn sich der Inhalt einer ADR ändert, bevor das Team die ADR akzeptiert, sollte der Eigentümer diese Änderungen genehmigen.

Das folgende Diagramm veranschaulicht den Prozess der Erstellung, Inhaberschaft und Einführung von ADR.



Nachdem das Team eine architektonische Entscheidung und deren Eigentümer identifiziert hat, stellt der ADR-Eigentümer das ADR im Vorgeschlagen Zustand zu Beginn des Prozesses bereit. ADRs im Vorgeschlagen Zustand sind zur Überprüfung bereit.

Der ADR-Eigentümer leitet dann den Überprüfungsprozess für das ADR ein. Ziel des ADR-Überprüfungsprozesses ist es, zu entscheiden, ob das Team das ADR akzeptiert, feststellt, dass es überarbeitet werden muss, oder ob es das ADR ablehnt. Das Projektteam, einschließlich des

Eigentümers, überprüft das ADR. Das Überprüfungstreffen sollte mit einem speziellen Zeitfenster für die Lektüre des ADR beginnen. Im Durchschnitt sollten 10 bis 15 Minuten ausreichen. Während dieser Zeit liest jedes Teammitglied das Dokument und fügt Kommentare und Fragen hinzu, um auf unklare Themen hinzuweisen. Nach der Überprüfungsphase liest der ADR-Verantwortliche jeden Kommentar vor und bespricht ihn mit dem Team.

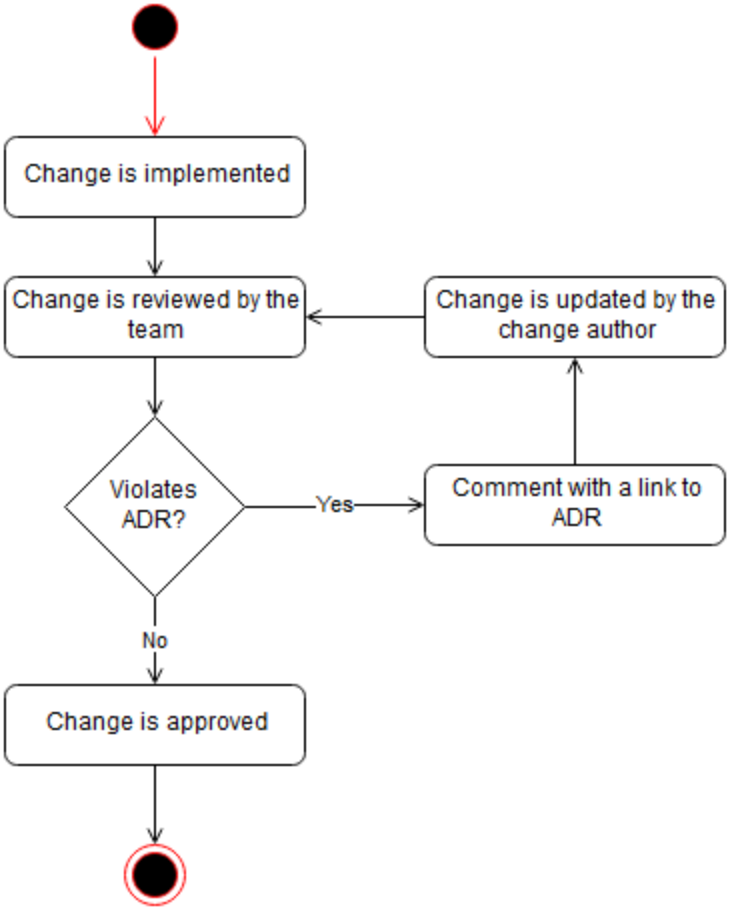
Findet das Team Maßnahmen zur Verbesserung der alternativen Streitbeilegung, bleibt der Status Vorgeschlagen des ADRs bestehen. Der ADR-Verantwortliche formuliert die Aktionen und fügt in Zusammenarbeit mit dem Team jeder Aktion einen Beauftragten hinzu. Jedes Teammitglied kann einen Beitrag leisten und die Aktionspunkte lösen. Es liegt in der Verantwortung des ADR-Eigentümers, den Überprüfungsprozess neu zu planen.

Das Team kann auch beschließen, das ADR abzulehnen. In diesem Fall fügt der ADR-Eigentümer einen Grund für die Ablehnung hinzu, um zukünftige Diskussionen zum gleichen Thema zu verhindern. Der Eigentümer ändert den ADR-Status in Abgelehnt.

Wenn das Team den ADR genehmigt, fügt der Eigentümer einen Zeitstempel, eine Version und eine Liste der Stakeholder hinzu. Der Besitzer aktualisiert dann den Status auf Akzeptiert.

ADRs und das von ihnen erstellte Entscheidungsprotokoll stellen die vom Team getroffenen Entscheidungen dar und bieten eine Historie aller Entscheidungen. Das Team verwendet die ADRs nach Möglichkeit als Referenz bei Code- und Architekturprüfungen. Neben der Durchführung von Codeprüfungen, Entwurfs- und Implementierungsaufgaben sollten die Teammitglieder ADRs bei strategischen Entscheidungen für das Produkt heranziehen.

Das folgende Diagramm zeigt den Prozess der Anwendung eines ADR zur Überprüfung, ob eine Änderung an einer Softwarekomponente den vereinbarten Entscheidungen entspricht.

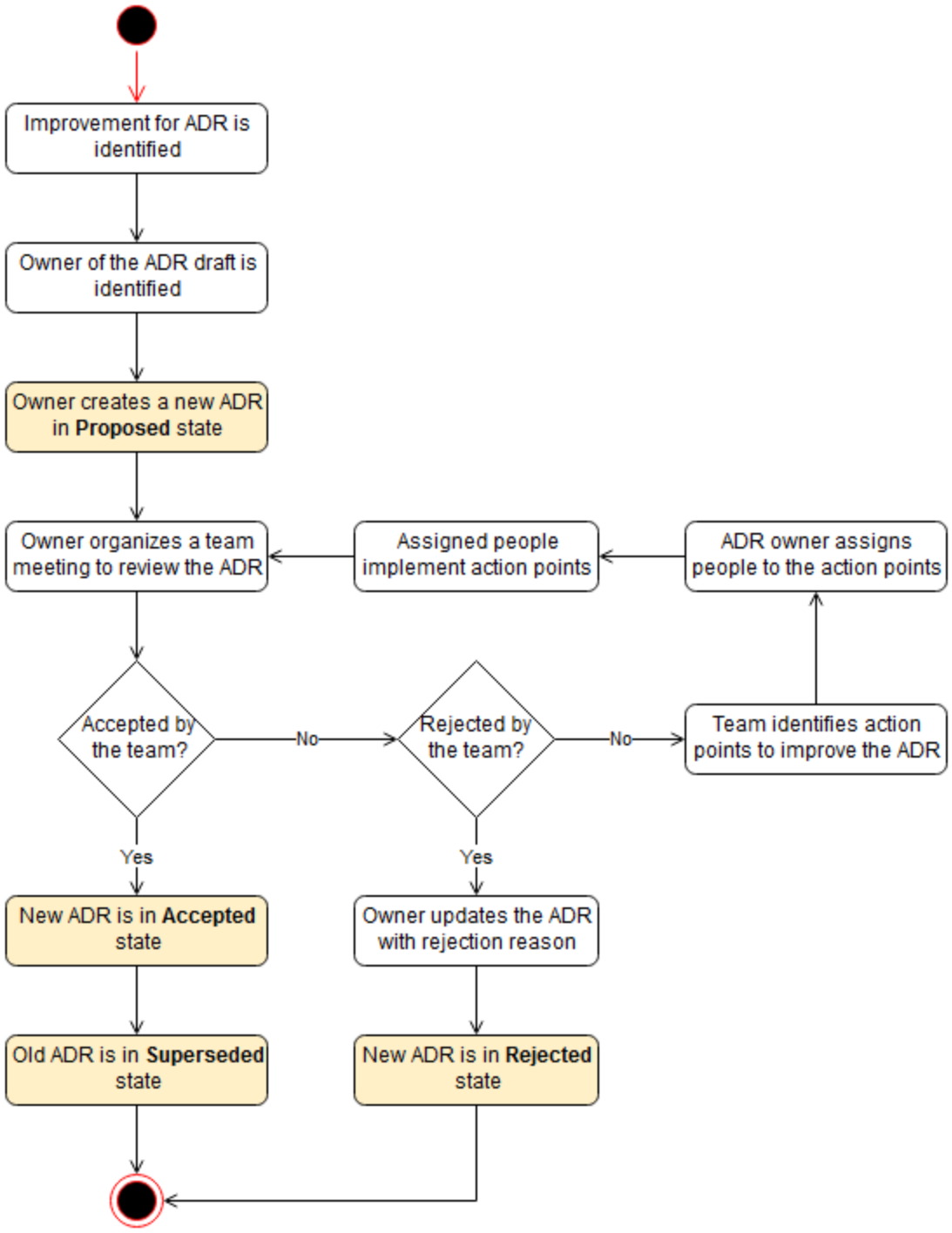


Es hat sich bewährt, dass jede Softwareänderung einem Peer-Review-Verfahren unterzogen werden sollte und mindestens eine Genehmigung erfordert. Während der Codeüberprüfung kann ein Code-Reviewer Änderungen feststellen, die gegen eine oder mehrere ADRs verstoßen. In diesem Fall bittet der Prüfer den Autor der Codeänderung, den Code zu aktualisieren, und teilt ihm einen Link zum ADR mit. Wenn der Autor den Code aktualisiert, wird er von Fachgutachtern genehmigt und mit der Hauptcodebasis zusammengeführt.

ADR-Überprüfungsverfahren

Das Team sollte ADRs als unveränderliche Dokumente behandeln, nachdem das Team sie akzeptiert oder abgelehnt hat. Änderungen an einem bestehenden ADR erfordern die Erstellung eines neuen ADR, die Einrichtung eines Überprüfungsprozesses für das neue ADR und die Genehmigung des

ADR. Wenn das Team das neue ADR genehmigt, sollte der Eigentümer den Status des alten ADR zu Ersetzt ändern. Das folgende Diagramm illustriert den Prozess.



Bewährte Methoden

Fördern Sie Eigentümerschaft. Jedes Mitglied des Projektteams sollte befugt sein, ein ADR zu erstellen und zu besitzen. Diese Praxis verteilt die architektonische Forschungsarbeit unter den Teammitgliedern und entlastet den Lösungsarchitekten oder Teamleiter von dieser Arbeit. Es fördert auch das Gefühl der Eigenverantwortung im Entscheidungsprozess. Dies hilft dem Team, diese Entscheidungen schneller zu treffen, anstatt sie als Entscheidungen zu behandeln, die von höheren Ebenen der Organisation auferlegt wurden.

Bewahren Sie den ADR-Verlauf auf. ADRs sollte eine Änderungshistorie haben, und jede Änderung sollte einen Besitzer haben. Wenn der ADR-Eigentümer das ADR aktualisiert, sollte er den Status des alten ADR auf Ersetzt ändern, seine Änderungen in der Änderungshistorie des neuen ADR vermerken und das alte ADR im Entscheidungsprotokoll aufbewahren.

Vereinbaren Sie regelmäßige Überprüfungstreffen. Wenn Sie an einem neuen Projekt (auf der grünen Wiese) arbeiten, kann der ADR-Prozess am Anfang sehr intensiv sein. Wir empfehlen Ihnen, regelmäßige ADR-Diskussions- und Überprüfungssitzungen vor oder nach dem täglichen Standup zu etablieren. Mit diesem Ansatz stabilisieren ADRs sich die definierten Werte innerhalb von zwei oder drei Sprints, und Sie können mit weniger Besprechungen ein solides Fundament aufbauen.

An einem ADRs zentralen Ort lagern. Jedes Projektmitglied sollte Zugriff auf die Sammlung von haben ADRs. Wir empfehlen Ihnen, sie ADRs an einem zentralen Ort zu speichern und auf der Hauptseite Ihrer Projektdokumentation darauf zu verweisen. Es gibt zwei beliebte Speicheroptionen ADRs:

- Ein Git-Repository, das die Versionierung erleichtert ADRs
- Eine Wiki-Seite, die das ADRs für alle Teammitglieder zugänglich macht

Adressieren Sie nicht konforme Codes. Das ADR-Verfahren löst das Problem des nicht konformen Legacy-Codes nicht. Wenn Sie über älteren Code verfügen, der den etablierten Code nicht unterstützt ADRs, können Sie entweder die veraltete Codebasis oder die veralteten Artefakte schrittweise aktualisieren und dabei neue Änderungen einführen, oder Ihr Team kann beschließen, den Code explizit umzugestalten, indem es technische Aufgaben erstellt.

Häufig gestellte Fragen

Was sind die Vorteile eines ADR-Verfahrens?

Das Projektteam sollte einen ADR-Prozess entwickeln, um architektonische Entscheidungen zu rationalisieren, wiederholte Diskussionen über dieselben architektonischen Themen zu verhindern und architektonische Entscheidungen effektiv zu kommunizieren.

Wann sollte das Projektteam ein ADR erstellen?

Das Projektteam sollte ein ADR für jeden Aspekt der Software erstellen, der sich auf die Struktur (Muster wie Microservices), nicht funktionale Anforderungen (Sicherheit, Hochverfügbarkeit und Fehlertoleranz), Abhängigkeiten (Kopplung von Komponenten), Schnittstellen (APIsund veröffentlichte Verträge) und Konstruktionstechniken (Bibliotheken, Frameworks, Tools und Prozesse) auswirkt.

Wie oft sollte das Projektteam ein ADR überprüfen?

Das Projektteam sollte das ADR mindestens einmal überprüfen, bevor es es akzeptiert.

Wer sollte ein ADR erstellen?

Jedes Teammitglied kann ein ADR erstellen. Wir empfehlen Ihnen, ein Konzept der Eigenverantwortung für zu fördern. ADRs Ein Autor, der Eigentümer eines ADR ist, sollte den ADR-Inhalt aktiv pflegen und kommunizieren. Andere Teammitglieder können jederzeit zu einem ADR-Verfahren beitragen. Der ADR-Eigentümer sollte Änderungen an einem ADR genehmigen.

Welche Informationen sollte ein ADR enthalten?

Jede alternative Streitbeilegung muss mindestens den Kontext der Entscheidung, die Entscheidung selbst und die Konsequenzen der Entscheidung für das Projekt und seine Ergebnisse definieren. Im Kontext sollten mögliche Lösungen genannt werden, die das Team in Betracht gezogen hat. Es sollte auch alle relevanten Informationen zum Projekt, zum Kunden oder zum Technologie-Stack enthalten. In der Entscheidung muss klar und in unmissverständlicher Sprache angegeben

werden, für welche Lösung sich das Team entschieden hat. Vermeiden Sie es, Wörter wie „sollte“ zu verwenden, und formulieren Sie jede Entscheidung so, dass sie sagt: „Wir verwenden...“ oder „Das Team muss...“. Im Abschnitt mit den Konsequenzen sollten alle bekannten Kompromisse bei der Entscheidungsfindung erwähnt werden. Jeder ADR muss einen Status und ein Änderungsprotokoll haben, das das Änderungsdatum und die Person enthält, die für die Änderung verantwortlich ist.

Wo finde ich ADR-Vorlagen?

Es sind mehrere Versionen und Varianten von ADR-Vorlagen verfügbar. Eine öffentliche Sammlung häufig verwendeter ADR-Vorlagen finden Sie im GitHub ADR-Repository.

Nächste Schritte und Ressourcen

Wir empfehlen Ihnen, klein anzufangen und die ADRs Vorteile für Ihr Team zu erkennen. Wenn Sie an einem laufenden Projekt arbeiten, identifizieren Sie die nächste architektonische Änderung und wenden Sie das vorgeschlagene ADR-Verfahren an, um Ihr erstes ADR zu erstellen.

Ein weiterer Ausgangspunkt ist die Dokumentation Ihres gesamten Softwareentwicklungsprozesses mithilfe von ADRs. Oft basiert der Entwicklungsprozess auf implizitem Wissen, das das Team in keiner Dokumentation erfasst hat. Die Dokumentation dieses Prozesses ermöglicht neuen Teammitgliedern ein reibungsloseres Erlebnis.

Wenn Sie noch ein Greenfield-Projekt haben, wenden Sie das ADR-Verfahren an und fangen Sie an, alle Entscheidungen von Anfang an in wenigen Sätzen festzuhalten. Sie können diese dann wiederholen ADRs und mit neuen Informationen ergänzen. Nachdem Sie Ihre erstellt haben ADRs, können Sie sie als Referenz in Ihrem Code-Review-Prozess verwenden.

Ressourcen

- Aufzeichnungen über Architekturentscheidungen. https://adr.github.io/.
- Mark Richards und Neal Ford. 2021. Grundlagen der Softwarearchitektur. Sewastopol: O'Reilly Media.

Anhang: Beispiel ADR

Titel

Diese Entscheidung definiert den Ansatz für den Lebenszyklus der Softwareentwicklung für die ABC-Anwendungsentwicklung.

Status

Accepted (Akzeptiert)

Date (Datum)

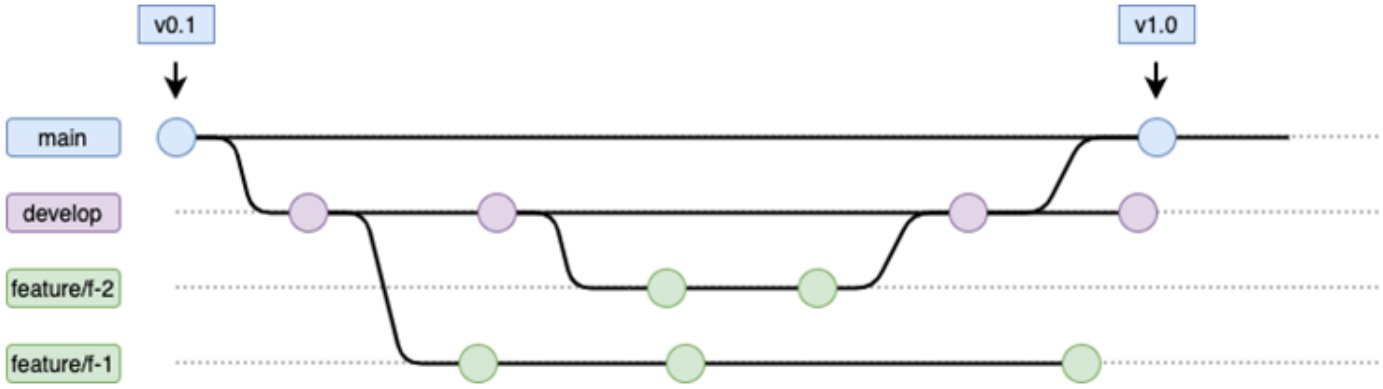
11.03.2022

Kontext

Bei der ABC-Anwendung handelt es sich um eine Paketlösung, die mithilfe eines Bereitstellungspakets in der Kundenumgebung bereitgestellt wird. Wir benötigen einen Entwicklungsprozess, der es uns ermöglicht, ein kontrollierbares Feature, einen Hotfix und eine Release-Pipeline einzurichten.

Entscheidung

Wir verwenden eine angepasste Version des GitFlowWorkflows, um die ABC-Anwendung zu entwickeln.



Der Einfachheit halber werden wir die `hotfix/*`- und `release/*`-Verzweigungen nicht verwenden, da die ABC-Anwendung in einem Paket zusammengefasst wird, anstatt in einer bestimmten

Umgebung bereitgestellt zu werden. Aus diesem Grund ist keine zusätzliche Komplexität erforderlich, die uns daran hindern könnte, schnell zu reagieren, um Fehler in Produktionsversionen oder Testversionen in einer separaten Umgebung zu beheben.

Im Folgenden ist die vereinbarte Verzeweigungsstrategie aufgeführt:

- Jedes Repository muss über einen geschützten `main`-Zweig verfügen , der zum Tagging von Veröffentlichungen verwendet wird.
- Jedes Repository muss für alle laufenden Entwicklungsarbeiten über einen geschützten `develop`-Zweig verfügen.

Konsequenzen

Positiv:

- Ein angepasster GitFlow Prozess wird es uns ermöglichen, die Versionsverwaltung der ABC-Anwendung zu kontrollieren.

Negativ:

- GitFlow ist komplizierter als Trunk-basierte Entwicklung oder GitHub Flow und hat mehr Aufwand.

Compliance

- Die `main`- und `develop`-Zweige in jedem Repository müssen als `Protected` gekennzeichnet sein.
- Änderungen von `main`- und `develop`-Zweige müssen mithilfe von Merge-Anforderungen weitergegeben werden.
- Für jede Merge-Anforderung ist mindestens eine Genehmigung erforderlich.

Hinweise

- Autor: Jane Doe
- Version: 0.1
- Änderungsprotokoll:
  - 0.1: Erste vorgeschlagene Version

Dokumentverlauf

In der folgenden Tabelle werden wichtige Änderungen in diesem Leitfaden beschrieben. Um Benachrichtigungen über zukünftige Aktualisierungen zu erhalten, können Sie einen RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Erste Veröffentlichung	—	16. März 2022

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.