Utilisation d’enregistrements de décisions architecturales pour rationaliser la prise de décisions techniques dans le cadre d’un projet de développement logiciel

AWS Conseils prescriptifs

AWS Conseils prescriptifs: Utilisation d’enregistrements de décisions architecturales pour rationaliser la prise de décisions techniques dans le cadre d’un projet de développement logiciel

Table of Contents

Introduction 1
Résultats commerciaux ciblés 2
Processus ADR 3
Champ d'application du processus ADR 3
Contenu de l'ADR 4
Processus d'adoption d'ADR 4
Processus de révision d'ADR 7
Bonnes pratiques 9
FAQ 10
Quels sont les avantages liés à la création d'un processus d'ADR ? 10
Quand l'équipe de projet doit-elle créer un ADR ? 10
À quelle fréquence l'équipe de projet doit-elle revoir un ADR ? 10
Qui doit créer un ADR ? 10
Quelles informations doit contenir un ADR ? 10
Où puis-je trouver des modèles d'ADR ? 11
Prochaines étapes et ressources 12
Annexe : exemple d'ADR 13
Historique du document 15
..... xvi

Utilisation d'enregistrements de décisions architecturales pour rationaliser la prise de décisions techniques dans le cadre d'un projet de développement logiciel

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

Mars 2022 (historique du document)

Ce guide présente le processus d'enregistrement des décisions architecturales (ADR) pour les projets de génie logiciel. ADRs soutenez l'alignement des équipes, documentez les orientations stratégiques d'un projet ou d'un produit et réduisez les efforts de prise de décision récurrents et fastidieux.

Au cours du développement de projets et de produits, les équipes d'ingénierie logicielle doivent prendre des décisions architecturales pour atteindre leurs objectifs. Ces décisions peuvent être techniques, comme le choix d'utiliser le modèle de ségrégation des responsabilités des requêtes de commande (CQRS), ou liées au processus, comme la décision d'utiliser le GitFlow flux de travail pour gérer le code source. La prise de ces décisions est un processus complexe et fastidieux. Les équipes doivent justifier, documenter et communiquer ces décisions aux parties prenantes concernées.

Trois principaux anti-modèles apparaissent souvent lors de la prise de décisions architecturales :

- Aucune décision n'est prise, par peur de faire le mauvais choix.
- Une décision est prise sans aucune justification, et l'on ne comprend pas pourquoi elle a été prise. Il en résulte que le même sujet est discuté plusieurs fois.
- La décision n'étant pas capturée dans un référentiel de décisions architecturales, les membres de l'équipe l'oublient ou ne savent pas qu'elle a été prise.

Il est particulièrement important d'aborder ces anti-modèles lors du processus de développement d'un produit ou d'un projet.

La capture de la décision, du contexte et des considérations qui ont conduit à la décision sous la forme d'un ADR permet aux parties prenantes actuelles et futures de collecter des informations sur les décisions prises et le processus de réflexion sous-jacent à chaque décision. Cela réduit le temps de développement de logiciels et fournit une meilleure documentation aux futures équipes.

Résultats commerciaux ciblés

ADRs cibler trois résultats commerciaux :

- Ils alignent les membres actuels et futurs de l'équipe.
- Ils définissent une orientation stratégique pour le projet ou le produit.
- Ils évitent les anti-modèles décisionnels en définissant un processus permettant de documenter et de communiquer correctement les décisions architecturales.

ADRs saisir le contexte de la décision pour informer les futures parties prenantes. Une collection de documents ADRs fournissant une expérience de transfert et de référence. Les membres de l'équipe ou du projet utilisent la collection d'ADR pour les projets de suivi et la planification des fonctionnalités du produit. La capacité de référence ADRs réduit le temps nécessaire au développement, aux révisions et aux décisions architecturales. ADRs permettent également aux autres équipes de tirer des leçons et de mieux comprendre les considérations prises par d'autres équipes de développement de projets et de produits.

Processus ADR

Un enregistrement de décision architecturale (ADR) est un document qui décrit un choix fait par l'équipe concernant un aspect important de l'architecture logicielle qu'elle envisage de générer. Chaque ADR décrit la décision architecturale, son contexte et ses conséquences. Les ADR ont des états et suivent donc un cycle de vie. Pour obtenir un exemple d'ADR, consultez l'annexe.

Le processus ADR produit une collection d'enregistrements de décisions architecturales. Cette collection crée le journal des décisions. Le journal des décisions fournit le contexte du projet ainsi que des informations détaillées sur l'implémentation et la conception. Les membres du projet parcourent les titres de chaque ADR pour avoir une vue d'ensemble du contexte du projet. Ils lisent les ADR pour approfondir les implémentations des projets et les choix de conception.

Lorsque l'équipe accepte un ADR, celui-ci devient immuable. Si de nouvelles connaissances nécessitent une décision différente, l'équipe propose un nouvel ADR. Lorsque l'équipe accepte le nouvel ADR, celui-ci remplace l'ADR précédent.

Champ d'application du processus ADR

Les membres du projet doivent créer un ADR pour chaque décision importante sur le plan architectural qui affecte le projet ou le produit logiciel, y compris les suivantes (Richards et Ford 2020) :

- Structure (par exemple, des modèles tels que les microservices)
- Non-functional exigences (sécurité, haute disponibilité et tolérance aux pannes)
- Dépendances (couplage de composants)
- Interfaces (API et contrats publiés)
- Techniques de construction (bibliothèques, cadres, outils et processus)

Les exigences fonctionnelles et non fonctionnelles sont les entrées les plus courantes du processus ADR.

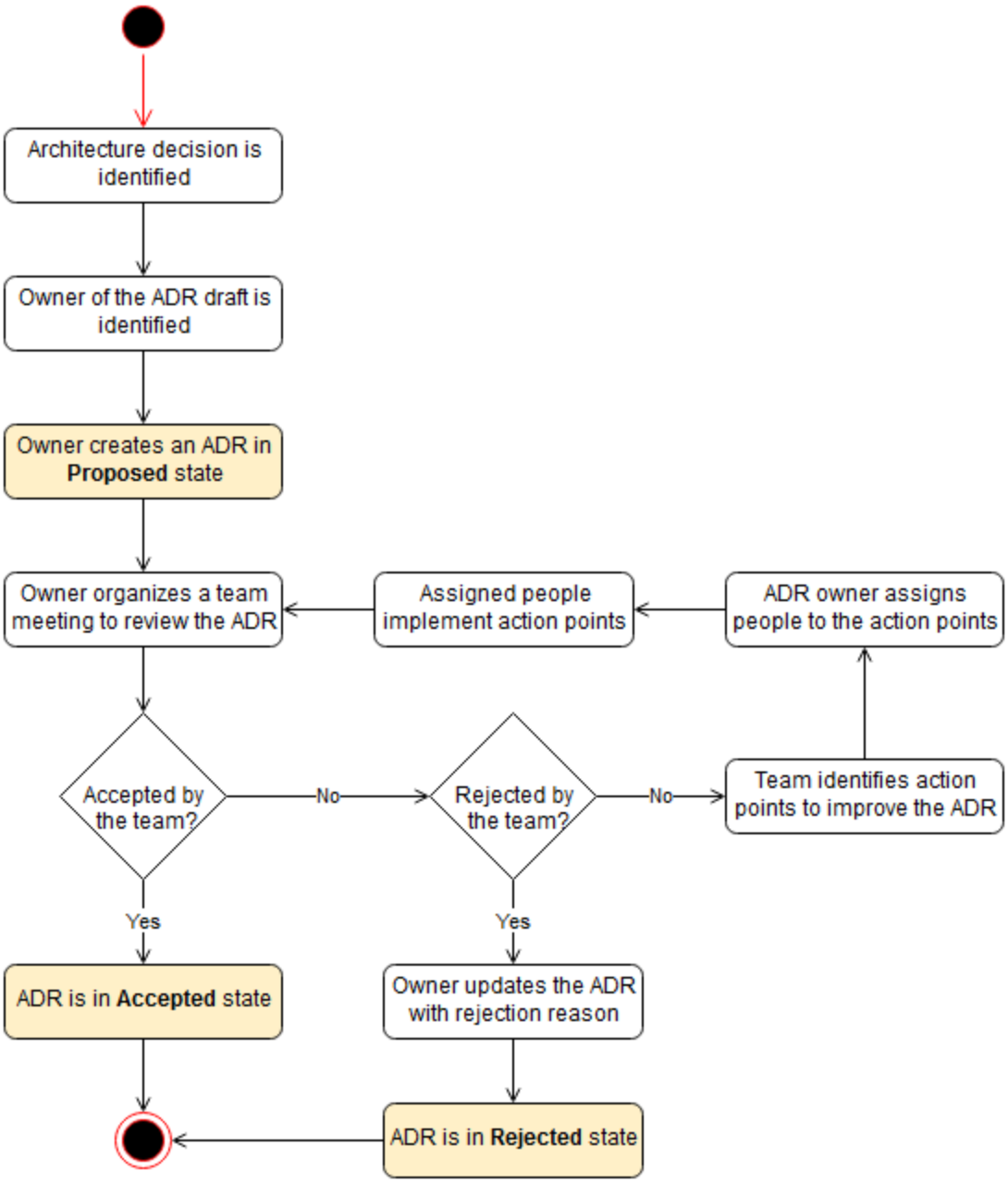
Contenu de l'ADR

Lorsque l'équipe identifie le besoin d'un ADR, un membre de l'équipe commence à rédiger l'ADR sur la base d'un modèle à l'échelle du projet. (Voir l' GitHuborganisation ADR pour des exemples de modèles.) Le modèle simplifie la création d'ADR et garantit que l'ADR capture toutes les informations pertinentes. Chaque ADR doit au minimum définir le contexte de la décision, la décision elle-même et les conséquences de la décision pour le projet et ses livrables. (Pour des exemples de ces sections, consultez l'annexe.) L'un des aspects les plus notables de la structure d'ADR est qu'elle se concentre sur le motif de la décision plutôt que sur la manière dont l'équipe l'a implémentée. Comprendre pourquoi l'équipe a pris cette décision permet aux autres membres de l'équipe de l'adopter plus facilement et empêche les autres architectes qui n'ont pas participé au processus décisionnel de revenir sur cette décision à l'avenir.

Processus d'adoption d'ADR

Chaque membre de l'équipe peut créer un ADR, mais l'équipe doit établir une définition de la propriété d'un ADR. Chaque auteur propriétaire d'un ADR doit activement maintenir et communiquer le contenu de l'ADR. Pour clarifier cette propriété, ce guide fait référence aux auteurs d'ADR comme des propriétaires d'ADR dans les sections suivantes. Les autres membres de l'équipe peuvent toujours contribuer à un ADR. Si le contenu d'un ADR change avant que l'équipe n'accepte l'ADR, le propriétaire doit approuver ces modifications.

Le schéma suivant illustre le processus de création, de propriété et d'adoption d'ADR.



Une fois que l'équipe a identifié une décision architecturale et son propriétaire, le propriétaire de l'ADR fournit l'ADR à l'état Proposé au début du processus. Les ADR à l'état Proposé sont prêts à être examinés.

Le propriétaire de l'ADR lance ensuite le processus d'examen de l'ADR. L'objectif du processus d'examen de l'ADR est de déterminer si l'équipe accepte l'ADR, si l'ADR doit être retravaillé ou si elle le rejette. L'équipe du projet, y compris le propriétaire, examine l'ADR. La réunion d'examen

doit commencer par un créneau horaire dédié à la lecture de l'ADR. En moyenne, 10 à 15 minutes devraient suffire. Pendant ce temps, chaque membre de l'équipe lit le document et ajoute des commentaires et des questions pour signaler les sujets flous. Après la phase d'examen, le propriétaire de l'ADR lit et discute de chaque commentaire avec l'équipe.

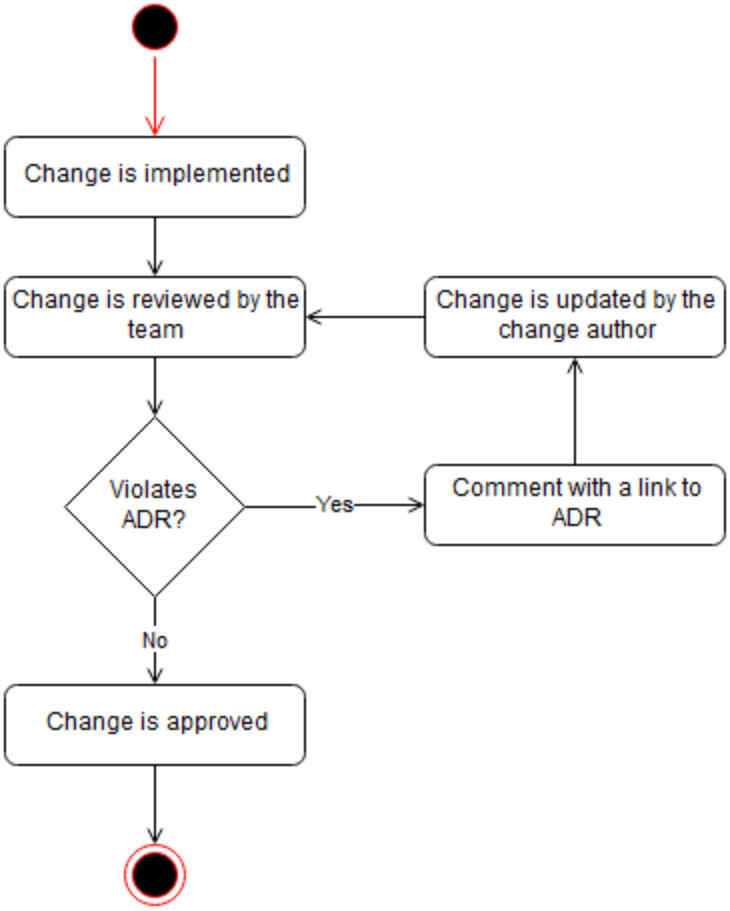
Si l'équipe trouve des points d'action pour améliorer l'ADR, l'état de l'ADR reste Proposé. Le propriétaire de l'ADR formule les actions et, en collaboration avec l'équipe, ajoute un destinataire à chaque action. Chaque membre de l'équipe peut apporter sa contribution et résoudre les points d'action. Il est de la responsabilité du propriétaire de l'ADR de replanifier le processus de révision.

L'équipe peut également décider de rejeter l'ADR. Dans ce cas, le propriétaire de l'ADR ajoute un motif de rejet afin d'éviter de futures discussions sur le même sujet. Le propriétaire change l'état de l'ADR en Rejeté.

Si l'équipe approuve l'ADR, le propriétaire ajoute un horodatage, une version et une liste des parties prenantes. Le propriétaire met ensuite à jour l'état en Accepté.

Les ADR et le journal des décisions qu'ils créent représentent les décisions prises par l'équipe et fournissent un historique de toutes les décisions. L'équipe utilise les ADR comme référence lors des examens du code et de l'architecture dans la mesure du possible. Outre les examens du code, les tâches de conception et les tâches d'implémentation, les membres de l'équipe doivent consulter les ADR pour prendre des décisions stratégiques concernant le produit.

Le schéma suivant montre le processus d'application d'un ADR pour valider si une modification apportée à un composant logiciel est conforme aux décisions convenues.

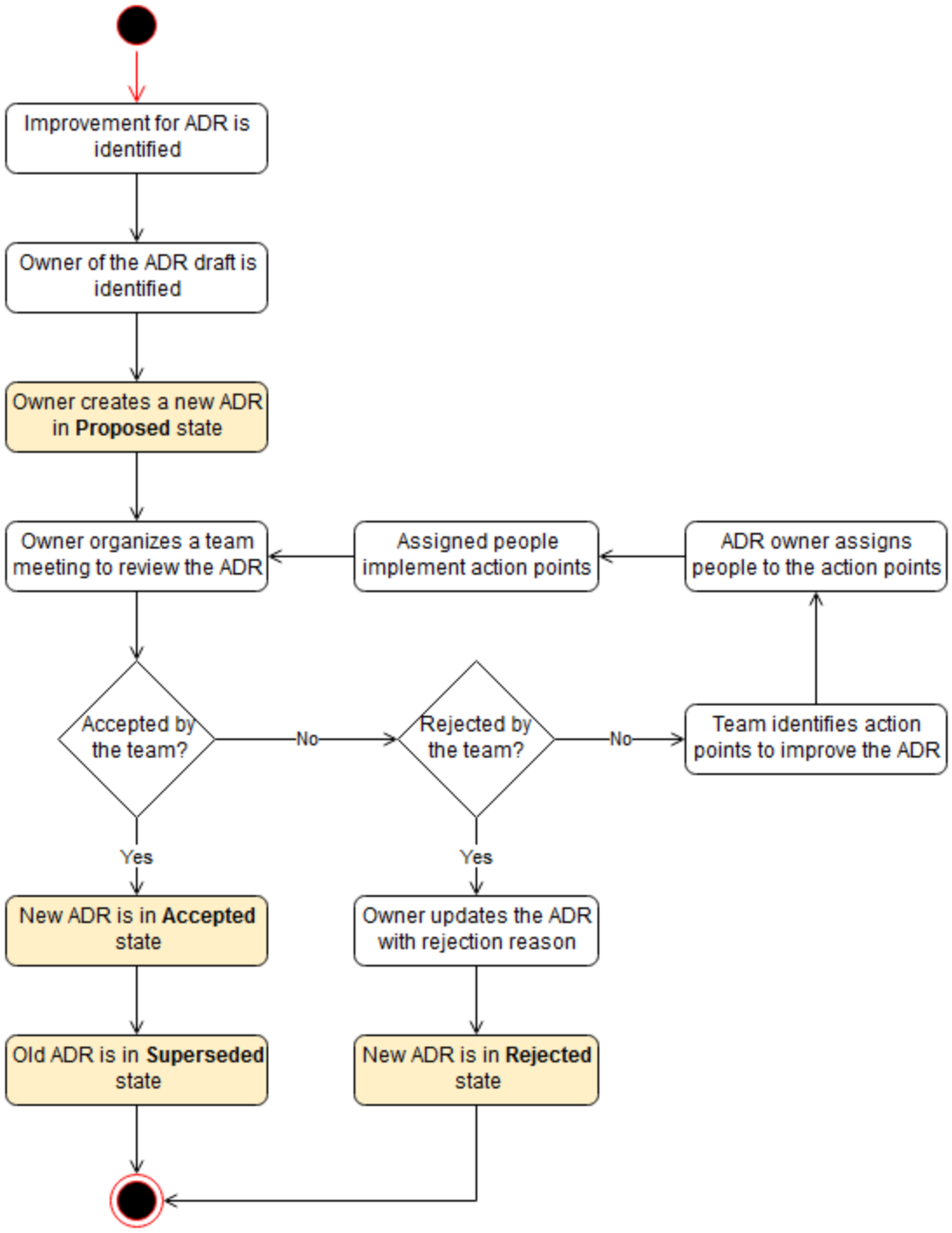


Il est recommandé que chaque modification logicielle fasse l'objet d'un examen par les pairs et nécessite au moins une approbation. Au cours de l'examen du code, un réviseur de code peut détecter des modifications qui enfreignent un ou plusieurs ADR. Dans ce cas, le réviseur demande à l'auteur du changement de code de le mettre à jour et partage un lien vers l'ADR. Lorsque l'auteur met à jour le code, celui-ci est approuvé par des réviseurs pairs et intégré à la base de code principale.

Processus de révision d'ADR

L'équipe doit traiter les ADR comme des documents immuables une fois qu'elle les a acceptés ou rejetés. Les modifications apportées à un ADR existant nécessitent la création d'un ADR, l'établissement d'un processus d'examen du nouvel ADR et l'approbation de l'ADR. Si l'équipe

approuve le nouvel ADR, le propriétaire doit modifier l'état de l'ancien ADR en Remplacé. Le schéma suivant illustre le processus de mise à jour.



Bonnes pratiques

Promouvoir la propriété. Chaque membre de l'équipe de projet doit être habilité à créer et à posséder un ADR. Cette pratique répartit le travail de recherche architecturale entre les membres de l'équipe et décharge le travail de l'architecte de solutions ou du chef d'équipe. Elle favorise également le sentiment d'appartenance dans le processus décisionnel. Cela permet à l'équipe d'adopter ces décisions plus rapidement au lieu de les traiter comme des décisions imposées par les niveaux supérieurs de l'organisation.

Préservez l'historique des ADR. ADRs doit avoir un historique des modifications, et chaque modification doit avoir un propriétaire. Lorsque le propriétaire de l'ADR le met à jour, il doit modifier l'état de l'ancien ADR en Remplacé, noter ses modifications dans l'historique des modifications du nouvel ADR et conserver l'ancien ADR dans le journal des décisions.

Planifier des réunions d'examen régulières. Si vous travaillez sur un nouveau projet (inédit), le processus ADR peut être assez intense au début. Nous vous recommandons d'établir une cadence de discussions régulières sur les ADR et de tenir des réunions d'examen avant ou après la présentation quotidienne. Avec cette approche, le résultat défini se ADRs stabilisera en deux ou trois sprints, et vous pourrez établir une base solide en réduisant le nombre de réunions.

Stockez ADRs dans un emplacement central. Chaque membre du projet doit avoir accès à la collection de ADRs. Nous vous recommandons de les stocker ADRs dans un emplacement central et de les référencer sur la page principale de la documentation de votre projet. Il existe deux options populaires pour le stockage ADRs :

- Un dépôt Git, qui facilite la création de versions ADRs
- Une page wiki, qui rend les informations ADRs accessibles à tous les membres de l'équipe

Traiter le code non conforme. Le processus ADR ne résout pas le problème de non-conformité du code existant. Si votre ancien code ne prend pas en charge le code établi ADRs, vous pouvez soit mettre à jour la base de code obsolète ou les artefacts progressivement, tout en introduisant de nouvelles modifications, soit votre équipe peut décider de refactoriser le code de manière explicite en créant des tâches de dette technique.

FAQ

Quels sont les avantages liés à la création d'un processus d'ADR ?

L'équipe du projet doit créer un processus d'ADR pour rationaliser la prise de décision architecturale, éviter les discussions répétées sur les mêmes sujets architecturaux et communiquer efficacement les décisions architecturales.

Quand l'équipe de projet doit-elle créer un ADR ?

L'équipe du projet doit créer un ADR pour chaque aspect du logiciel qui affecte la structure (modèles tels que les microservices), les exigences non fonctionnelles (sécurité, haute disponibilité et tolérance aux pannes), les dépendances (couplage de composants), les interfaces (APIset contrats publiés) et les techniques de construction (bibliothèques, cadres, outils et processus).

À quelle fréquence l'équipe de projet doit-elle revoir un ADR ?

L'équipe du projet doit revoir l'ADR au moins une fois avant de l'accepter.

Qui doit créer un ADR ?

Chaque membre de l'équipe peut créer un ADR. Nous vous recommandons de promouvoir une notion de propriété pour ADRs. Un auteur propriétaire de l'ADR doit activement maintenir et communiquer le contenu de l'ADR. Les autres membres de l'équipe peuvent toujours contribuer à un ADR. Le propriétaire de l'ADR doit approuver les modifications apportées à un ADR.

Quelles informations doit contenir un ADR ?

Chaque ADR doit au minimum définir le contexte de la décision, la décision elle-même et les conséquences de la décision pour le projet et ses livrables. Le contexte doit mentionner les solutions possibles envisagées par l'équipe. Il doit également contenir toutes les informations pertinentes relatives au projet, au client ou à la pile technologique. La décision doit clairement indiquer, dans un langage impératif, la solution que l'équipe a décidé d'adopter. Évitez d'utiliser des mots tels que « devrait » et formulez chaque décision comme « Nous utilisons… » ou « L'équipe doit utiliser… ». La

section sur les conséquences doit mentionner tous les compromis connus liés à la prise de décision. Chaque ADR doit avoir un statut et un journal des modifications contenant la date de modification et le nom de la personne responsable de la modification.

Où puis-je trouver des modèles d'ADR ?

Plusieurs versions et variantes de modèles d'ADR sont disponibles. Pour une collection publique de modèles ADR couramment utilisés, consultez le GitHub référentiel ADR.

Prochaines étapes et ressources

Nous vous recommandons de commencer modestement et de constater les avantages que cela ADRs apporte à votre équipe. Si vous travaillez sur un projet en cours, identifiez le prochain changement architectural et appliquez le processus d'ADR proposé pour créer votre premier ADR.

Un autre point de départ consiste à documenter l'ensemble de votre processus de développement logiciel en utilisant ADRs. Souvent, le processus de développement repose sur des connaissances tacites que l'équipe n'a consignées dans aucune documentation. La documentation de ce processus offre une expérience plus fluide pour les nouveaux membres de l'équipe.

Si vous travaillez sur un projet inédit, appliquez le processus ADR et commencez à capturer toutes les décisions depuis le début en quelques phrases. Vous pouvez ensuite les répéter ADRs et les compléter avec de nouvelles informations. Après avoir établi votre ADRs, vous pouvez commencer à les utiliser comme référence dans votre processus de révision du code.

Ressources

- Architecture Decision Records. https://adr.github.io/.
- Mark Richards et Neal Ford. 2021. Fundamentals of Software Architecture. Sébastopol : O'Reilly Media.

Annexe : exemple d'ADR

Titre

Cette décision définit l'approche du cycle de vie du développement logiciel pour le développement d'applications ABC.

Statut

Acceptée

Date

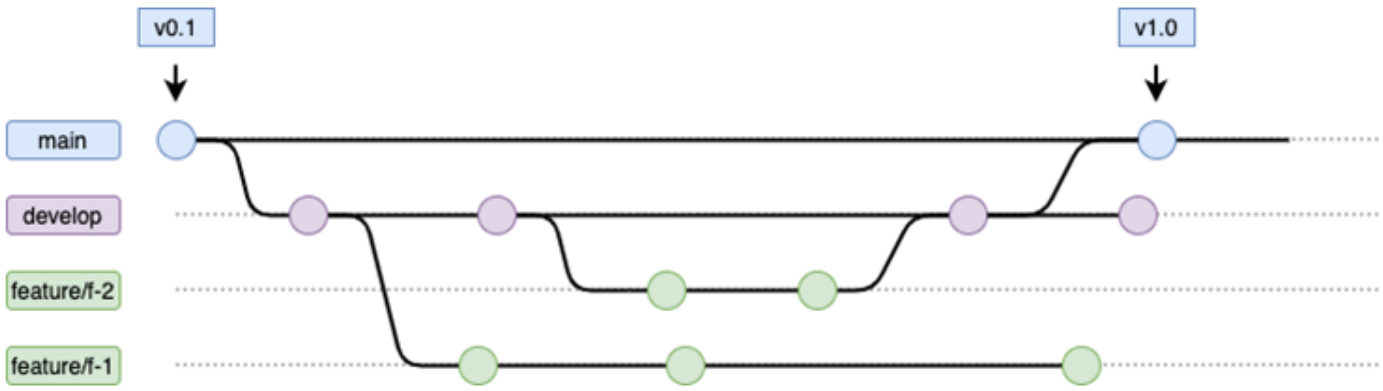
03/03/11

Contexte

L'application ABC est une solution packagée qui sera déployée dans l'environnement du client à l'aide d'un package de déploiement. Nous avons besoin d'un processus de développement qui nous permettra de disposer d'une fonctionnalité, d'un correctif et d'un pipeline de versions contrôlables.

Décision

Nous utilisons une version adaptée du GitFlowflux de travail pour développer l'application ABC.



Pour des raisons de simplicité, nous n'utiliserons pas les branches `hotfix/*` et `release/*`, car l'application ABC sera packagée au lieu d'être déployée dans un environnement spécifique. C'est pourquoi il n'est pas nécessaire d'ajouter une complexité supplémentaire qui pourrait nous empêcher

de réagir rapidement pour corriger les bogues dans les versions de production, ou de tester les versions dans un environnement distinct.

La stratégie de ramification convenue est la suivante :

- Chaque référentiel doit disposer d'une branche `main` protégée qui sera utilisée pour étiqueter les versions.
- Chaque référentiel doit disposer d'une branche `develop` protégée pour tous les travaux de développement en cours.

Conséquences

Positives :

- Un GitFlow processus adapté nous permettra de contrôler le versionnement des versions de l'application ABC.

Négatives :

- GitFlow est plus complexe que le développement ou le GitHub flux basés sur des troncs et entraîne une charge plus importante.

Conformité

- Les branches `main` et `develop` de chaque référentiel doivent être marquées comme `Protected`.
- Les modifications apportées aux branches `main` et `develop` doivent être propagées à l'aide de demandes de fusion.
- Au moins une approbation est requise pour chaque demande de fusion.

Remarques

- Auteur : Jane Doe
- Version : 0.1
- Journal des modifications :
  - 0.1 : version initiale proposée

Historique du document

Le tableau suivant décrit les modifications importantes apportées à ce guide. Pour être averti des mises à jour à venir, abonnez-vous à un fil RSS.

Modification	Description	Date
Publication initiale	—	16 mars 2022

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.