Using architectural decision records to streamline technical decision-making for a software development project

AWS Linee guida prescrittive

AWS Linee guida prescrittive: Using architectural decision records to streamline technical decision-making for a software development project

Table of Contents

Utilizzo dei record sulle decisioni architetturali per semplificare il processo decisionale tecnico per un progetto di sviluppo software

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

Marzo 2022 (cronologia dei documenti)

Questa guida introduce il processo ADR (Architectural Decision Records) per i progetti di ingegneria del software. ADRs supporta l'allineamento del team, documenta le direzioni strategiche per un progetto o un prodotto e riduci gli sforzi decisionali ricorrenti e dispendiosi in termini di tempo.

Durante lo sviluppo di progetti e prodotti, i team di ingegneria del software devono prendere decisioni in materia di progettazione architetturale per raggiungere i propri obiettivi. Queste decisioni possono essere tecniche, ad esempio decidere di utilizzare il modello CQRS (Command Query Responsibility Segregation), o legate al processo, come decidere di utilizzare il flusso di lavoro per gestire il codice sorgente. GitFlow L'elaborazione di queste decisioni rappresenta un processo lungo e complesso, dal momento che i team devono giustificarle, documentarle e comunicarle alle parti interessate.

Durante l'elaborazione di tali decisioni architetturali, spesso emergono tre anti-pattern principali:

- Non si prende alcuna decisione, per paura di fare la scelta sbagliata.
- Non viene fornita alcuna spiegazione alla decisione adottata, motivo per cui non se ne comprendono i motivi. Questo fa sì che lo stesso argomento venga discusso più volte.
- La decisione non viene archiviata in un repository delle decisioni architetturali, quindi i membri del team non vengono informati dell'adozione di una decisione o se ne dimenticano.

È particolarmente importante affrontare questi anti-pattern durante il processo di sviluppo di un prodotto o di un progetto.

La documentazione, sotto forma di ADR, della decisione, del contesto e delle considerazioni che hanno portato ad adottarla consente agli stakeholder attuali e futuri di raccogliere informazioni sulle decisioni prese e sulle relative riflessioni. Ciò riduce i tempi di sviluppo del software e fornisce una documentazione migliore per i team futuri.

Obiettivi aziendali specifici

ADRs mirano a tre risultati aziendali:

- Allineano i membri del team attuali e futuri.
- Stabiliscono una direzione strategica per il progetto o il prodotto.
- Evitano gli anti-pattern decisionali tramite la definizione di un processo per documentare e comunicare correttamente le decisioni architetturali.

ADRs cogliere il contesto della decisione per informare le future parti interessate. Una raccolta di ADRs informazioni sull'esperienza di trasferimento e sulla documentazione di riferimento. I membri del team o del progetto utilizzano la raccolta ADR per i progetti successivi e per la pianificazione delle funzionalità del prodotto. La possibilità di fare riferimento ADRs riduce il tempo necessario durante lo sviluppo, le revisioni e le decisioni architettoniche. ADRs consenti inoltre agli altri team di imparare e ottenere informazioni approfondite sulle considerazioni fatte da altri team di sviluppo di progetti e prodotti.

Processo di ADR

Un record della decisione sull'architetttura (ADR, architectural decision record) è un documento che descrive una scelta fatta dal team su un aspetto significativo dell'architettura software che intende creare. Ogni ADR descrive la decisione architettonica, il suo contesto e le sue conseguenze. Gli ADR presentano degli stati e quindi seguono un ciclo di vita. Per un esempio di ADR, consulta l'appendice.

Il processo di ADR produce una raccolta di record delle decisioni sull'architettura. Questa raccolta forma il log delle decisioni. Il log delle decisioni fornisce il contesto del progetto e informazioni dettagliate sull'implementazione e sulla progettazione. I membri del progetto sfogliano i titoli di ogni ADR per avere una panoramica del contesto del progetto. Leggono gli ADR per approfondire le implementazioni del progetto e le scelte di progettazione.

Quando il team accetta un ADR, questo diventa non modificabile. Se in base a nuove informazioni è richiesta una decisione diversa, il team propone un nuovo ADR. Quando il team accetta il nuovo ADR, questo sostituisce l'ADR precedente.

Ambito del processo di ADR

I membri del progetto devono creare un ADR per ogni decisione significativa dal punto di vista architettonico che influisca sul progetto o sul prodotto software, incluse le seguenti (Richards e Ford 2020):

- Struttura (ad esempio, pattern come i microservizi)
- Non-functional requisiti (sicurezza, alta disponibilità e tolleranza agli errori)
- Dipendenze (accoppiamento di componenti)
- Interfacce (API e contratti pubblicati)
- Tecniche di costruzione (librerie, framework, strumenti e processi)

I requisiti funzionali e non funzionali sono gli input più comuni del processo di ADR.

Contenuti dell'ADR

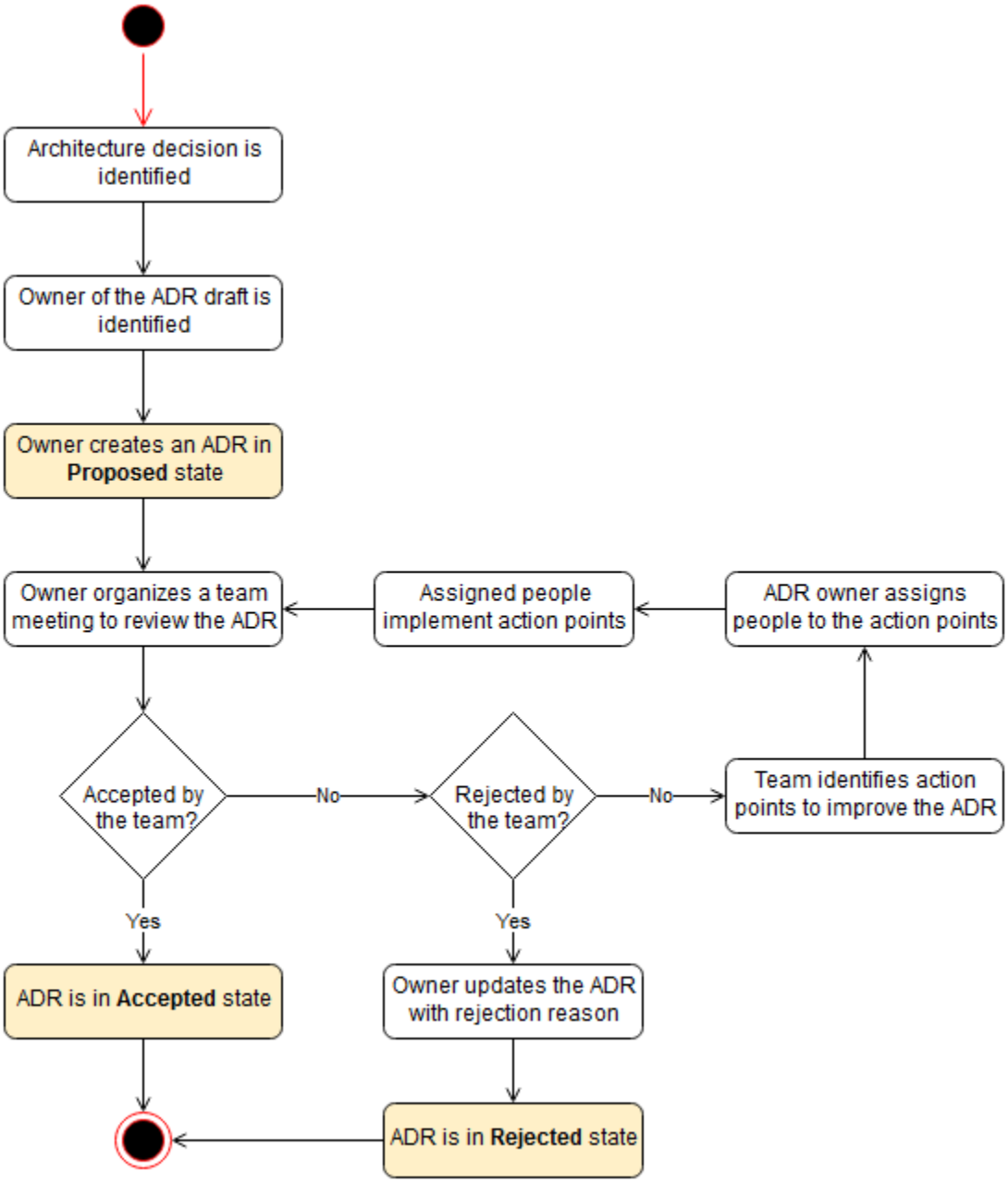
Quando il team identifica la necessità di un ADR, un membro del team inizia a scriverlo sulla base di un modello a livello di progetto. (Vedi l' GitHuborganizzazione ADR per esempi di modelli.) Il modello

semplifica la creazione di ADR e garantisce che l'ADR acquisisca tutte le informazioni pertinenti. Come minimo, ogni ADR dovrebbe definire il contesto della decisione, la decisione stessa e le conseguenze della decisione sul progetto e sui suoi risultati. (Per alcuni esempi di queste sezioni, consulta l'appendice). Uno degli aspetti più importanti della struttura dell'ADR è che si concentra sul motivo della decisione piuttosto che sul modo in cui il team l'ha implementata. Capire perché il team ha preso la decisione rende più facile adottarla per gli altri membri del team e impedisce ad altri architetti che non sono stati coinvolti nel processo decisionale di annullare tale decisione in futuro.

Processo di adozione dell'ADR

Ogni membro del team può creare un ADR, ma il team deve stabilire una definizione di proprietà per l'ADR. Ogni autore proprietario di un ADR dovrebbe mantenere e comunicare attivamente i contenuti dell'ADR. Per spiegare in modo chiaro questa proprietà, nelle seguenti sezioni questa guida si riferisce agli autori dell'ADR come a Proprietari dell'ADR. Gli altri membri del team possono sempre contribuire a un ADR. Se il contenuto di un ADR cambia prima che il team lo accetti, il proprietario deve approvare tali modifiche.

Il seguente diagramma illustra il processo di creazione, proprietà e adozione dell'ADR.



Dopo che il team ha identificato una decisione architettonica e il suo proprietario, il proprietario dell'ADR fornisce l'ADR nello stato Proposto all'inizio del processo. Gli ADR nello stato Proposto sono pronti per la revisione.

Il proprietario dell'ADR avvia quindi il processo di revisione dell'ADR. L'obiettivo del processo di revisione dell'ADR è decidere se il team lo accetta, se stabilisce che necessita di una rielaborazione o se lo rifiuta. Il team del progetto, incluso il proprietario, esamina l'ADR. La riunione di revisione

dovrebbe iniziare con un intervallo di tempo dedicato alla lettura dell'ADR. In media, dovrebbero essere sufficienti dai 10 ai 15 minuti. Durante questo periodo, ogni membro del team legge il documento e aggiunge commenti e domande per segnalare gli argomenti poco chiari. Dopo la fase di revisione, il proprietario dell'ADR legge ad alta voce e discute ogni commento con il team.

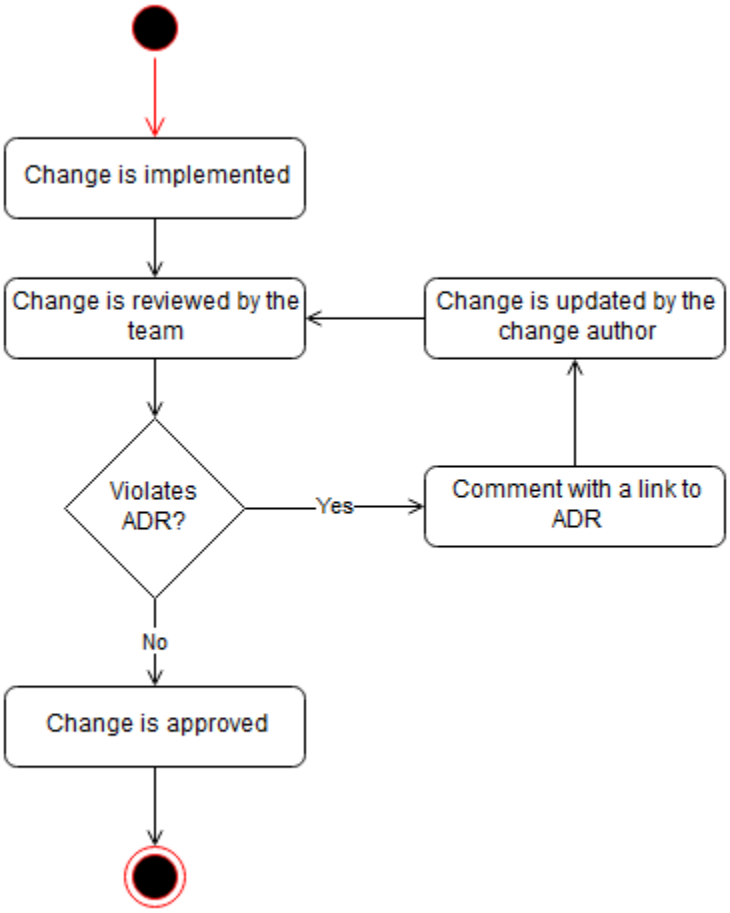
Se il team trova punti su cui intervenire per migliorarlo, l'ADR rimane nello stato Proposto. Il proprietario dell'ADR formula le azioni e, in collaborazione con il resto del team, assegna ciascuna azione a un membro del team. Ogni membro del team può contribuire e risolvere i punti di intervento. È responsabilità del proprietario dell'ADR riprogrammare il processo di revisione.

Il team può anche decidere di rifiutare l'ADR. In questo caso, il proprietario dell'ADR aggiunge un motivo per il rifiuto in modo da evitare future discussioni sullo stesso argomento. Il proprietario modifica lo stato dell'ADR in Respinto.

Se il team approva l'ADR, il proprietario aggiunge un timestamp, una versione e un elenco delle parti interessate. Il proprietario modifica quindi lo stato in Accettato.

Gli ADR e il log delle decisioni creati rappresentano le decisioni prese dal team e forniscono una cronologia di tutte le decisioni. Il team utilizza gli ADR come riferimento durante le revisioni del codice e dell'architettura, ove possibile. Oltre a eseguire revisioni del codice e attività di progettazione e implementazione, i membri del team dovrebbero consultare gli ADR per le decisioni strategiche relative al prodotto.

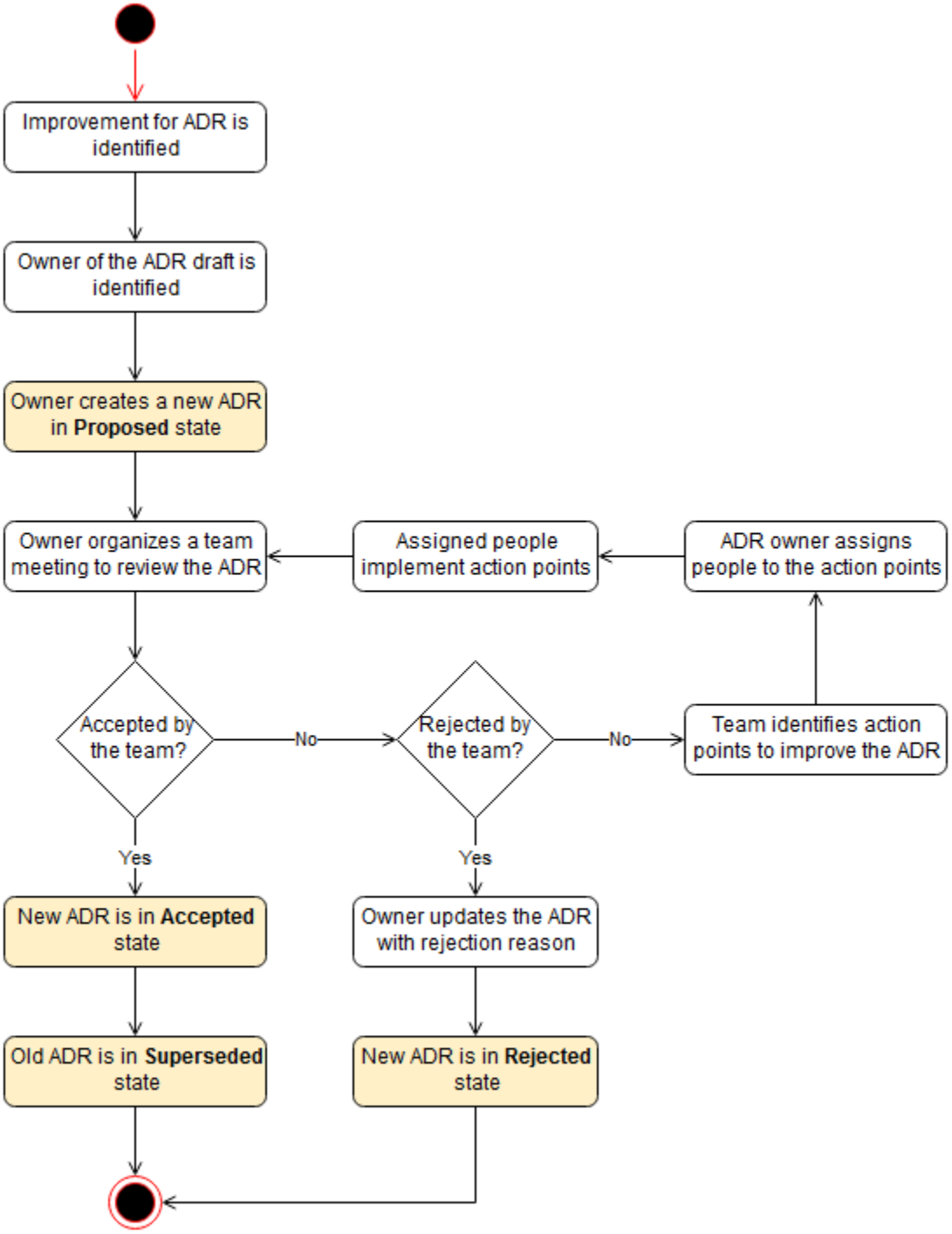
Il seguente diagramma mostra il processo di applicazione di un ADR per verificare se una modifica in un componente software è conforme alle decisioni concordate.



È buona norma che ogni modifica del software sia sottoposta a revisione tra pari e richieda almeno un'approvazione. Durante la revisione del codice, un revisore potrebbe trovare modifiche che violano uno o più ADR. In tal caso, il revisore chiede all'autore della modifica del codice di aggiornare il codice e condivide un link all'ADR. Quando l'autore aggiorna il codice, questo viene approvato dai revisori e inserito nella base di codice principale.

Processo di revisione dell'ADR

Dopo averli accettati o rifiutati, il team deve trattare gli ADR come documenti non modificabili. Le modifiche a un ADR esistente richiedono la creazione di un nuovo ADR, l'istituzione di un processo di revisione per il nuovo ADR e l'approvazione dello stesso. Se il team approva il nuovo ADR, il proprietario del vecchio ADR deve modificarne lo stato in Sostituito. Il seguente diagramma illustra il processo di aggiornamento.


Improvement for ADR is identified
Owner of the ADR draft is identified
Owner creates a new ADR in **Proposed** state
Owner organizes a team meeting to review the ADR
Assigned people implement action points
ADR owner assigns people to the action points
Accepted by the team?
No
Rejected by the team?
No
Team identifies action points to improve the ADR
Yes
Yes
New ADR is in **Accepted** state
Owner updates the ADR with rejection reason
Old ADR is in **Superseded** state
New ADR is in **Rejected** state

Best practice

Promuovere la responsabilizzazione. Ogni membro del team di progetto dovrebbe essere incoraggiato a creare e possedere un ADR (Architectural Design Record, record di progettazione architetturale). Questo approccio consente di distribuire il lavoro di ricerca architetturale tra i membri del team, riducendo il carico del progettista di soluzioni o del responsabile del team. Inoltre, favorisce il coinvolgimento nel processo decisionale, consentendo al team di adottare rapidamente tali soluzioni invece di considerarle come decisioni imposte dai livelli superiori dell'organizzazione.

Conserva la cronologia ADR. ADRs dovrebbe avere una cronologia delle modifiche e ogni modifica dovrebbe avere un proprietario. Quando il proprietario dell'ADR lo aggiorna, deve modificare lo stato del vecchio ADR in Sostituito, annotare le modifiche nella relativa cronologia del nuovo ADR e mantenere il vecchio ADR nel log delle decisioni.

Pianificare incontri di revisione periodici. Il processo ADR può essere piuttosto intenso all'inizio, se si tratta di un nuovo progetto (greenfield). Ti consigliamo di impostare una cadenza di riunioni regolari in modo da discutere e revisionare gli ADR prima o dopo le attività giornaliere. Con questo approccio, la definizione ADRs si stabilizzerà in due o tre sprint e potrai costruire una solida base con un minor numero di riunioni.

ADRs Conserva in una posizione centrale. Ogni membro del progetto dovrebbe avere accesso alla raccolta di ADRs. Ti consigliamo di archiviarli ADRs in una posizione centrale e di consultarli nella pagina principale della documentazione del progetto. Esistono due opzioni popolari per l'archiviazione ADRs:

- Un repository Git, che semplifica la versione ADRs
- Una pagina wiki, che lo rende ADRs accessibile a tutti i membri del team

Affrontare il problema relativo alla non conformità del codice legacy. Il processo ADR non risolve il problema relativo alla non conformità del codice legacy. Se disponi di un codice legacy che non supporta quanto stabilito ADRs, puoi aggiornare gradualmente la base di codice o gli artefatti obsoleti, introducendo nuove modifiche, oppure il tuo team può decidere di rifattorizzare il codice in modo esplicito creando attività tecniche inutili.

Domande frequenti

Quali sono i vantaggi della creazione di un processo ADR?

La creazione di un processo ADR da parte del team di progetto consente di semplificare il processo decisionale, evitare continue discussioni sugli stessi argomenti e comunicare in modo efficace le decisioni in materia di progettazione architetturale.

Quando è necessario creare un ADR?

Il team di progetto dovrebbe creare un ADR per ogni aspetto del software che influisce sulla struttura (modelli come i microservizi), sui requisiti non funzionali (sicurezza, alta disponibilità e tolleranza agli errori), sulle dipendenze (accoppiamento di componenti), sulle interfacce (APIse contratti pubblicati) e sulle tecniche di costruzione (librerie, framework, strumenti e processi).

Con quale frequenza il team di progetto deve esaminare un ADR?

Il team di progetto deve esaminare l'ADR almeno una volta prima di accettarlo.

Chi si occupa della creazione di un ADR?

Ogni membro del team può creare un ADR. Ti consigliamo di promuovere un concetto di proprietà per. ADRs L'autore che possiede l'ADR deve mantenere e comunicare attivamente i relativi contenuti. Gli altri membri del team possono sempre contribuire a un ADR. Il proprietario dell'ADR deve approvare le modifiche apportate a un ADR.

Quali informazioni deve contenere un ADR?

Come minimo, ogni ADR deve definire il contesto della decisione, la decisione stessa e le relative conseguenze sul progetto e sui suoi risultati. Il contesto deve indicare le possibili soluzioni prese in considerazione dal team. Deve inoltre contenere tutte le informazioni pertinenti relative al progetto, al cliente o allo stack tecnologico. La decisione deve indicare chiaramente la soluzione che il team ha deciso di adottare, con un linguaggio imperativo. Evita di usare termini come "dovrebbe" e formula ogni decisione con "utilizziamo..." o "il team deve usare...". La sezione relativa alle conseguenze deve

menzionare tutti i compromessi noti nell'adottare la decisione. Ogni ADR deve avere uno stato e un log delle modifiche con la data in cui è stata apportata la modifica e la persona che ne è responsabile.

Dove posso trovare i modelli ADR?

Sono disponibili diverse versioni e varianti dei modelli ADR. Per una raccolta pubblica di modelli ADR di uso comune, consultate l'archivio GitHub ADR.

Risorse e passaggi successivi

Ti consigliamo di iniziare in piccolo e vedere i ADRs vantaggi che ne derivano per il tuo team. Se stai lavorando a un progetto in corso, identifica la prossima modifica architetturale e applica il processo ADR proposto per creare il primo ADR.

Un altro punto di partenza è documentare l'intero processo di sviluppo del software utilizzando ADRs. Il processo di sviluppo si basa spesso su conoscenze tacite che il team non ha acquisito tramite raccolte di documenti. La documentazione di questo processo fornisce un'esperienza più fluida per i nuovi membri del team.

Se stai lavorando a un nuovo progetto (greenfield), applica il processo ADR e inizia a documentare tutte le decisioni fin dall'inizio, in poche frasi. È quindi possibile ripeterli ADRs e integrarli con nuove informazioni. Dopo aver stabilito le vostre ADRs, potete iniziare a usarle come riferimento nel processo di revisione del codice.

Risorse

- Record sulle decisioni architetturali: https://adr.github.io/.
- Mark Richards e Neal Ford, 2021. Fundamentals of Software Architecture. Sebastopol, O'Reilly Media.

Appendice: ADR di esempio

Titolo

Questa decisione definisce l'approccio al ciclo di vita dello sviluppo del software per l'implementazione di applicazioni ABC.

Stato

Accettato

Data

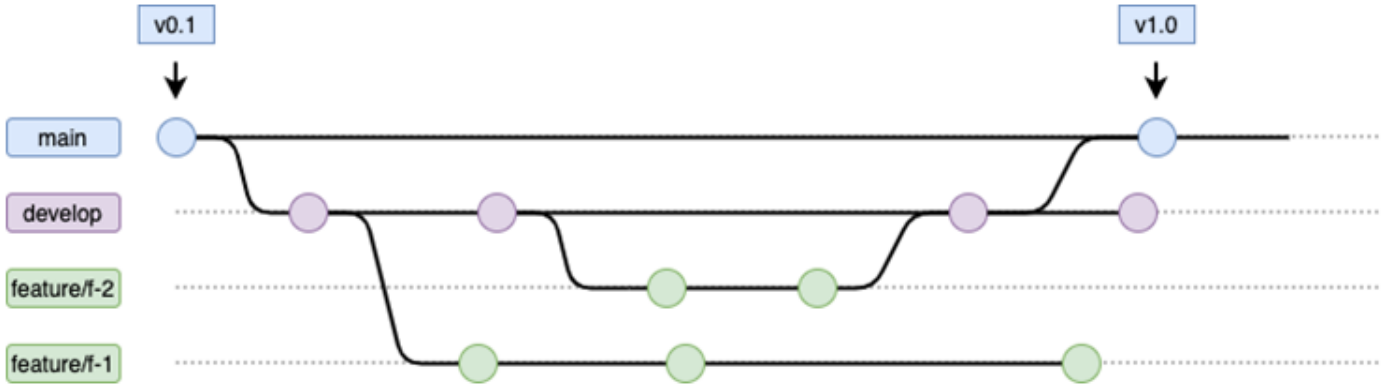
2022-03-11

Contesto

L'applicazione ABC è una soluzione pacchettizzata che verrà distribuita nell'ambiente del cliente tramite un pacchetto di implementazione. È necessario un processo di sviluppo che ci consenta di disporre di funzionalità controllabili, hotfix e pipeline di rilascio.

Decisione

Utilizziamo una versione adattata del GitFlowflusso di lavoro per sviluppare l'applicazione ABC.



Per semplicità non si utilizzeranno le ramificazioni `hotfix/*` e `release/*`, dal momento che l'applicazione ABC verrà assemblata e non implementata in un ambiente specifico. Per questo motivo, non è necessario introdurre nuovi livelli di complessità che potrebbero impedirci di reagire rapidamente alla correzione di bug nelle versioni di produzione o di testare le versioni in un ambiente separato.

Di seguito è riportata la strategia di ramificazione concordata:

- Ogni repository deve avere una ramificazione `main` protetta, da utilizzare per applicare tag alle versioni.
- Ogni repository deve avere una ramificazione `develop` protetta per tutti i lavori di sviluppo in corso.

Conseguenze

Aspetti positivi:

- GitFlow Il processo adattato ci consentirà di controllare il controllo delle versioni di rilascio dell'applicazione ABC.

Aspetti negativi:

- GitFlow è più complicato dello sviluppo o del GitHub flusso basato su trunk e ha maggiori costi generali.

Conformità

- Le ramificazioni `main` e `develop` in ogni repository devono essere contrassegnate come `Protected`.
- Le modifiche apportate alle ramificazioni `main` e `develop` devono essere propagate utilizzando le richieste di unione.
- È richiesta almeno un'approvazione per ogni richiesta di unione.

Note

- Autore: Jane Doe
- Versione: 0.1
- Changelog:
  - 0.1: versione iniziale proposta

Cronologia dei documenti

La tabella seguente descrive le modifiche significative apportate a questa guida. Per ricevere notifiche sugli aggiornamenti futuri, puoi abbonarti a un feed RSS.

Modifica	Descrizione	Data
Pubblicazione iniziale	—	16 marzo 2022

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.