

Softwarové architektury

Event Storming

Co je event storming?

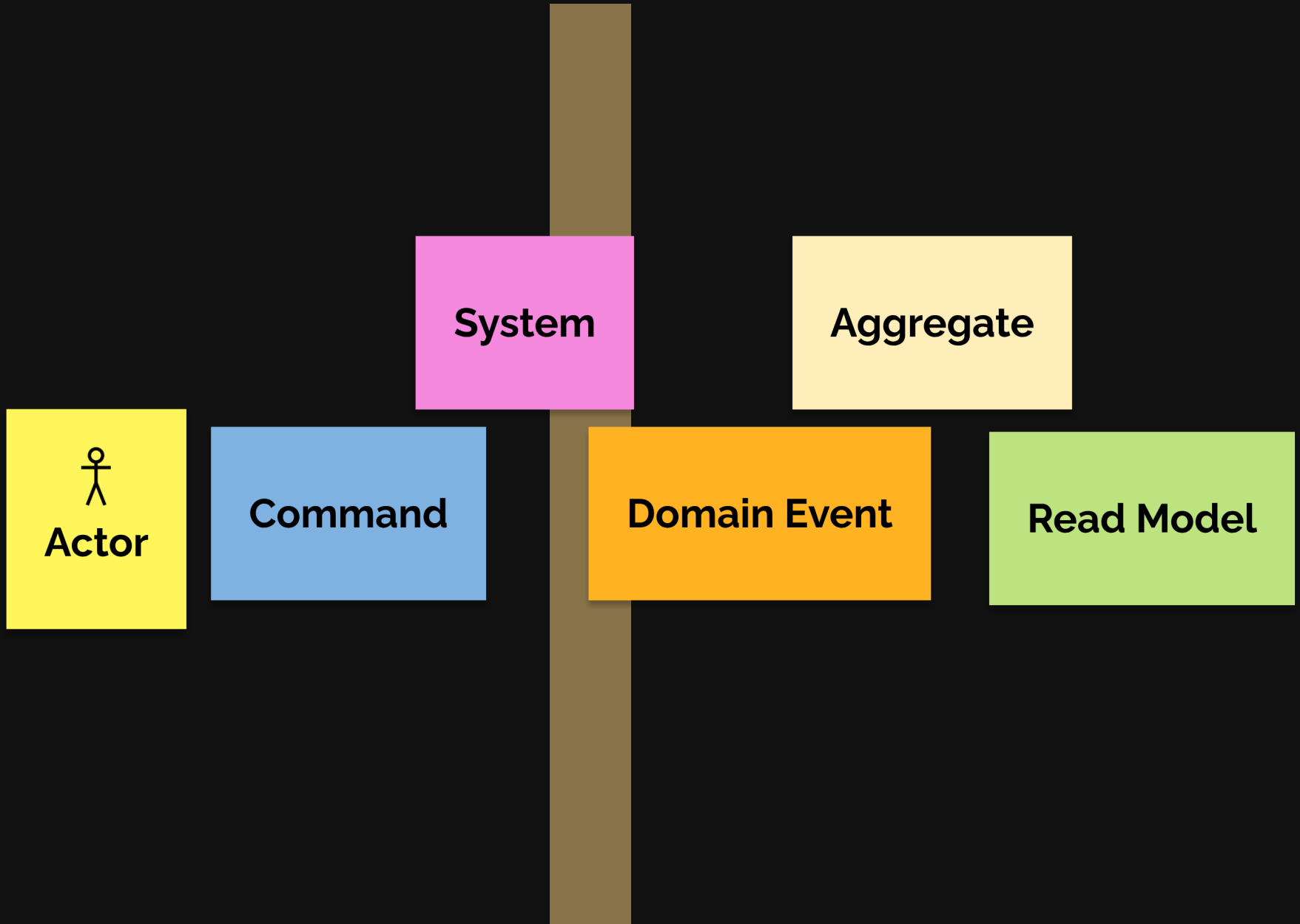
- kreativní technika pro modelování business procesů a systémů
- umožňuje rychlé a efektivní pochopení systému napříč vývojáři, doménovými experty, manažery
- vznikne pohled na systém poskytující přehled doménových pojmů (ubiquity language) a celého business workflow

Základní principy:

- Zaměření na časovou osu událostí (jaké události se dějí a v jakém pořadí)
- Vizualizace procesů pomocí barevných lepících lístků
- Iterativní přístup k návrhu a optimalizaci doménového modelu

**Cílem je společné
porozumění procesu a najít
úzká místa či příležitosti pro
zlepšení**

pivotal event



Proč Event Storming vznikl a kde se používá

Důvod vzniku: Potřeba lepší komunikace mezi technickými a netechnickými účastníky vývoje

Aplikace: V rámci Domain-Driven Design (DDD). Usnadňuje pochopení komplexních doménových modelů a procesů

Použití v agilních metodologiích:

- Při rychlé iteraci a přizpůsobování se změnám
- Vhodné pro inception fázi projektů

Jaké problémy řeší Event Storming

Komunikace mezi týmy:

- Spojuje vývojáře, doménové experty, manažery a další stakeholdery
 - Zajišťuje, že všechny strany rozumí stejným pojmům a událostem
-

Složité systémy

- Umožňuje vizualizovat složité systémy a identifikovat jejich klíčové komponenty
- Pomáhá odhalit nesrovnalosti a nedostatky v procesech

Doménové modelování:

- Podporuje vytvoření přehledného doménového modelu, který reflektuje skutečné procesy Umožňuje identifikovat vazby mezi různými částmi systému

Výhody a nevýhody Event Stormingu

Výhody:

- Rychlá pochopení procesu
- Zlepšení komunikace mezi týmy
- Přehledné doménové modelování
- Identifikace problémů

Nevýhody:

- Potřeba dobrého facilitátora
- Riziko chaosu při neorganizované facilitaci
- Může být časově náročné pro velké domény

Typy Event Stormingu

Big Picture Event Storming

Popis celkového systému nebo procesu

Process Level Event Storming

Detailní zobrazení jednoho procesu

Design Level Event Storming

Specifikace a návrh systému; vhodný pro vývojáře - jak konkrétně navrhnout funkce a integrace

Účastníci v Event Stormingu

Facilitátor/mediátor – role moderátora, schopnosti pracovat s lidmi a organizovat

Doménoví experti – jejich znalosti přispívají k celkovému obrazu, umí vysvětlit pojmy a

procesy Vývojáři a technici – propojení technické a doménové perspektivy

Stakeholdeři a manažeři – dávají pohled na proces, výstupy a očekávání

Základní komponenty Event Stormingu

Události (Events) - jsou klíčovým prvkem, definují, co se v systému stalo
Agregáty (Aggregates) - objekty, které tvoří relační rámec
Příkazy (Commandy)* - uživatelem vyvolané akce
Aktér (Actor) - ten kdo smí vyvolat command
Read modely/Query* - dotazy na data, která jsou potřeba jako vstup příkazu
Časová osa - události jsou seřazeny chronologicky, aby odhalily procesní tok

- CQRS - Command Query Request Segregation

Události

Událost (Event) = popis něčeho, co se stalo v systému.

Pro pojmenování používáme dokonavý vid/minulý čas.

Události jsou objektivní, reflektují skutečnost, ne plánování nebo hypotézy

Je to první věc, kterou modelujeme. Události pomáhají porozumět procesu. A jsou základní stavební blok pro další modelování systému

Command

Instrukce, kterou aktér vydává systému. Například volání API nebo kliknutí na tlačítko.
Tj. definují akce, které mají být v systému vykonány.

Příkazy vedou k událostem – každá událost je důsledkem nějakého příkazu (ikdyž mohou být i jiných událostí)

Mohou být označeny aktéry, kteří je vyvolávají.

Obsahují sloveso a podstatné jméno popisující operaci v přítomném čase.

Agregát

- seskupení událostí v rámci operace nebo entity (u event sourced systému) nebo objekty v transakčním rámci = reprezentace dat, která bude ukládat
- má jen jeden vstupní bod přes který pracujeme (aggregate root)

Časová osa – to podstatné!

- chronologické seřazení událostí, příkazů a agregátů. Jdeme zleva doprava (od prvním možných operací po následující až do uzavření flow)

Co budeme potřebovat?

Materiály a nástroje

Tabule nebo velká role papíru

- nutné pro vizualizaci celého procesu
- může být digitální (Miro, Figma)
- můžete klidně využít i dlouhou bílou zed'

Barevné lepící lístky

- Různé barvy pro různé typy prvků: události, příkazy, aktéři
- Umožňují snadnou vizuální identifikaci a organizaci

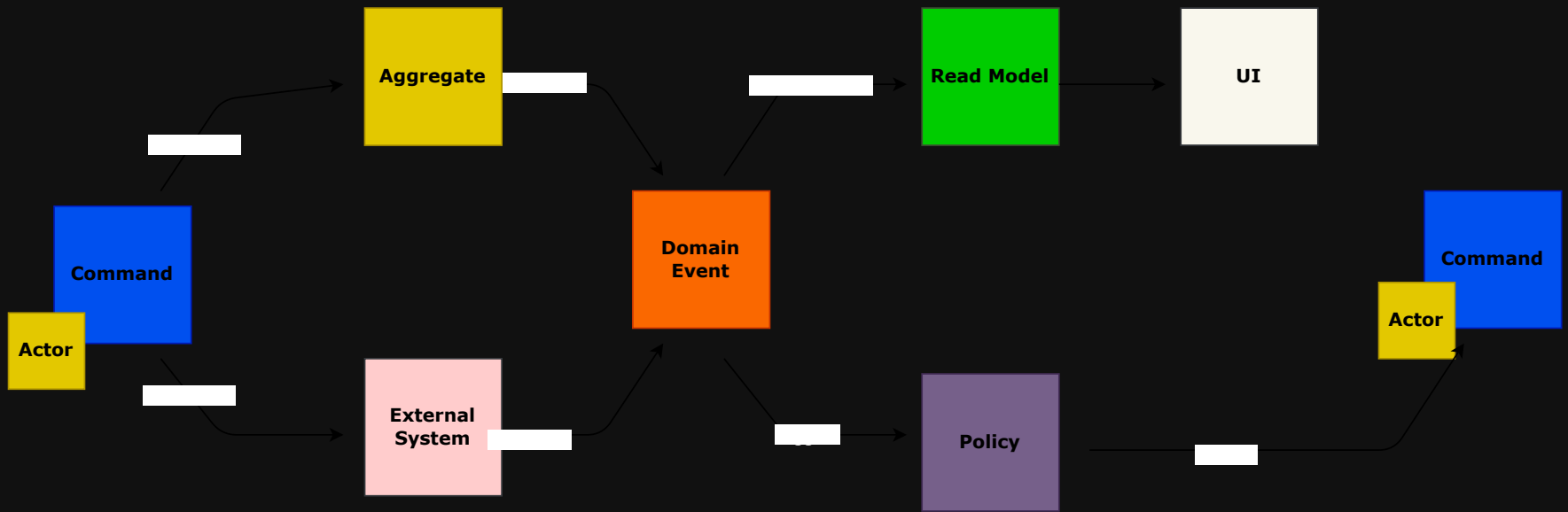
Fixy a psací potřeby

- Jasně a čitelné značení
- Různé barvy pro odlišení důležitých informací

Foták pro záložní záznam

Barvy lepících papírků

Událost (Event) – Oranžová Příkaz (Command) – Modrá Agregát – Žlutá Aktér (Actor) –
Žlutá, malý papírek Read model – Zelená UI - bílá Externí systém - Růžová Policy -
Fialová Issue - Červená



**Pokud si něčím nejsme jisti,
otočíme papírek o 45st**

Jak připravit vhodné prostředí

In-person

- Je možné pracovat online, ale není to tolik efektivní a produktivní

Dostatek prostoru

- Možnost pohybu kolem plátna pro různé účastníky

Pohodlné a neformální prostředí

- Stimulace kreativity a otevřené diskuze
- Umožňuje efektivní brainstorming

Časová dotace

- Vyhradte dostatek času pro meeting, obvykle několik hodin

Stanovení cíle

Cíl:

Každý meeting musí mít jasně definovaný cíl Např. návrh nového systému, zlepšení procesu

Klíčové otázky pro určení cíle:

- Co konkrétně chceme vyřešit?
- Jaké jsou hlavní problémy?
- Jaký je očekávaný výstup z tohoto sezení?

Zajištěte, aby všichni účastníci měli stejné očekávání a byli zaměřeni na stejný cíl

Průběh Event Stormingu

Event Storming sezení probíhá ve 3 hlavních fázích:

- Identifikace událostí
- Přidání příkazů, aktérů a agregátů
- Analýza a závěrečné diskuse

Cíl: Spolupracovat na vizualizaci systému a hledat příležitosti pro zlepšení

Fáze 1A – Identifikace událostí

Začněte od událostí (Events):

- Každý účastník přidává události (oranžové lístky) na tabuli/plátno
- Události popisují klíčové momenty, co se stalo v procesu nebo systému
- Bez omezení nebo hierarchie – jde o volný brainstorming

Tipy: Soustřeďte se na to, co se skutečně stalo, ne co by se mělo stát Diskutujte nad každou událostí a ujistěte se, že všichni rozumí jejímu významu

Fáze 1B - Seřazení událostí do časové osy

- Seřadíme události v pořadí, jak jdou za sebou
- Pokud je třeba přidáme další události

Fáze 2 – Přidání příkazů, aktérů a agregátů

Příkazy (Commands):

Přidejte modré lístky pro každou akci Diskutujte nad tím, jaký příkaz vedl k události
Může se stát, že budou chybět některé události. Pokud chybí, přidejte je.

Aktéři (Actors):

Identifikujte, kdo vydal příkazy, a označte je malými žlutými papírky. Určete, jak aktéři
interagují se systémem

Agregáty:

Mezi commandy a události přidejte agregáty na žlutém papírku

Read model (Zelené lístky):

Vytvořte, kde je potřeba zobrazení dat (např. reporty nebo dashboardy nebo vstupní data)

Fáze 3 – Analýza a závěrečná diskuse

Revize časové osy:

Projděte si události, příkazy a aktéry v pořadí, jakým probíhají Diskutujte o sekvencích a příčinných souvislostech

Hledání bottlenecků a problémů:

Identifikujte úzká místa, konflikty nebo nejasnosti Hledejte místa, kde jsou procesy pomalé, chybné nebo nejasné (otočení o 45 a červené papírky)

Zlepšení a změny:

Navrhňte, kde by mohlo dojít ke zlepšení procesu Diskutujte možné změny v návrhu nebo implementaci systému

Výstup:

- Vytvoření akčního plánu, další kroky, digitalizace

Doporučení

Facilitátor:

Zajistí, že se diskuse udržuje na správné cestě a zaměřuje se na cíle Povzbuzuje všechny účastníky k aktivnímu zapojení

Iterativní přístup:

Je běžné, že během sezení bude nutné některé prvky upravit a přidat. Jsou to papírky = rychle je vytvořím, přesunu a nebo zahodím. Event Storming je iterativní – proces se může několikrát zopakovat

Zaznamenání výsledků:

Po sezení zdokumentujte vizualizaci a klíčové poznatky Výstupy mohou sloužit jako podklady pro další fáze vývoje nebo analýzy

DEMO TIME

That's it.

See you next time.