

Reaktivní Manifest

Published on September 16 2014. (v2.0)

Organizace pracující napříč různými odvětvími vývoje software objevují nezávisle stále stejné vzory v návrhu a implementaci softwarových systémů. Takové systémy jsou pak robustnější, odolnější, pružnější a nabízejí řešení více odpovídající moderním požadavkům.

V poslední době dochází k těmto proměnám na základě rapidních změn požadavků na softwarová řešení. Před pouhými pár lety měla rozsáhlá aplikace jen deset serverů, odezvu v sekundách, hodiny offline provozu kvůli údržbě a gigabyty dat. Dnešní aplikace běží na všem od mobilních telefonů po cloudové cluster s tisíci multi-core procesory. Uživatelé očekávají odezvu v rámci milisekund a 100% dostupnost; data měříme v petabytech. Dnešní požadavky nelze zkrátka uspokojit včerejšími strukturami.

Věříme, že je třeba uceleného přístupu k návrhu systémů, a že všechny nezbytné aspekty byly již rozpoznány: systém musí být responzivní, odolný, pružný a řízený zprávami. Takový systém potom nazýváme reaktivní.

Systémy navržené jako reaktivní jsou pružnější, volně provázané a [škálovatelné](#). Díky tomu je jejich vývoj snazší a změny jednodušší. Takové systémy jsou tolerantnější k [výpadkům](#), které řeší elegantnějším způsobem, než kolapsem celého systému; mají vysokou odezvu a nabízejí [uživateli](#) interaktivní zpětnou vazbu.

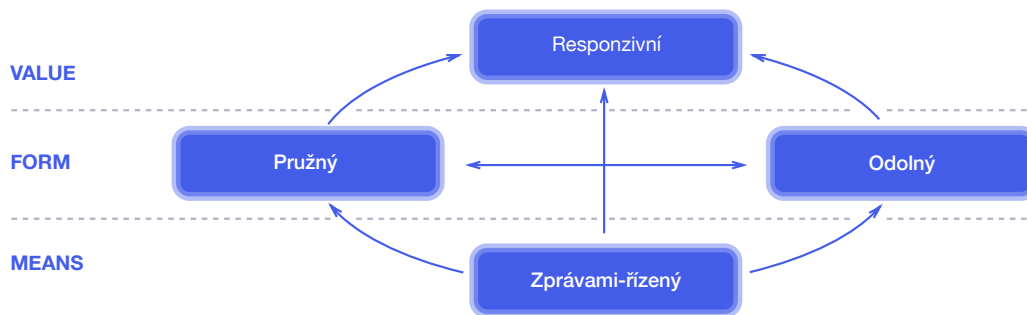
Reaktivní systém je:

Responzivní: [Systém](#) odpoví v požadovaném čase, pokud je nějaká odpověď vůbec možná. Rychlá odezva je nejen předpoklad funkčnosti a použitelnosti, ale umožňuje také rychlou a efektivní detekci problémů. Responzivní systémy poskytují rapidní konzistentní časovou odezvu zajišťující spolehlivost služby. Tato konzistence usnadňuje zpracování chyb a zvyšuje důvěru v kvalitu služby, což podporuje nové iterace vývoje.

Odolný: Systém zůstává responzivní i v případě [výpadku](#). Toto neplatí pouze pro kritické systémy s garantovanou vysokou dostupností - každý neodolný systém je po výpadku nedostupný. Odolnost je dosažena [replikací](#), zamezením šíření výpadku, [izolací](#) a [delegací](#) zpracování chyb. K výpadku může dojít u každé z [komponent](#) systému; izolací komponent lze dosáhnout stavu, kdy dojde v různých částech systému k výpadku, aniž by došlo k výpadku systému jako celku. Zotavení každé z komponent je delegováno na jinou (externí) komponentu a dostupnost je zajištěna replikací tam, kde je to nezbytné. Uživatel dané komponenty tak není zatížen nutností zpracovat její případné poruchy přímo.

Pružný: Systém zůstává responzivní i při proměnlivé zátěži. Reaktivní systém umí reagovat na změny zátěže v podobě proměnlivé frekvence požadavků pomocí zvyšování či snižování [zdrojů](#) přidělených dané službě. Pružný systém nemá žádné úzké hrdlo a umožňuje distribuci požadavků mezi komponentami pomocí sdílení nebo replikace. Reaktivní systémy podporují prediktivní škálovací algoritmy zohledňující míru zátěže v reálném čase. [Pružnost](#) je takto dosažitelná i na běžných nenákladných zařízeních a platformách.

Zprávami-řízený: Jasně oddělení komponent v reaktivním systému je založeno na [asynchronním zasílání zpráv](#), což zajišťuje volnou provázanost, izolaci a [transparentnost polohy](#). Toto oddělení také umožňuje reprezentaci a delegaci zpracování [výpadků](#) ve formě zpráv. Řízení zprávami dále umožňuje i správu zátěže, pružnost a řízení toku dat pomocí správy a monitorování front zpráv a aplikací [back-pressure](#) tam, kde třeba. Transparentnost polohy zpráv umožňuje komunikaci s komponentami pro kontrolu či správu výpadků bez nutnosti změny kódu nebo sémantiky programu nezávisle na tom, zda se komponenta nachází ve vzdáleném clusteru nebo na stejném stroji. [Neblokující](#) komunikace umožňuje efektivní využití [zdroje](#), protože komponenta zůstává aktivní i v době mezi příjmem zpráv, což vede k nižším režijním nákladům.



Rozsáhlé systémy se skládají z menších a závisí tedy na reaktivních vlastnostech svých komponent. To znamená, že reaktivní principy musí být aplikovány na všech úrovních abstrakce. Ty nejrozsáhlejší systémy na světě obsluhující miliardy uživatelů denně jsou na těchto principech postaveny. Je na čase aplikovat je vědomě hned od začátku vývoje namísto jejich objevování znovu a znovu.