MUNI
FI

Přednáška 8
Klíčové atributy kvality

CORE013 Vývoj softwarových systémů: od myšlenky k funkčnímu řešení

8. Klíčové atributy kvality

- Udržovatelnost
- Výkonnost
- Spolehlivost
- Bezpečnost
- Použitelnost

Domácí práce a příprava na příští přednášku

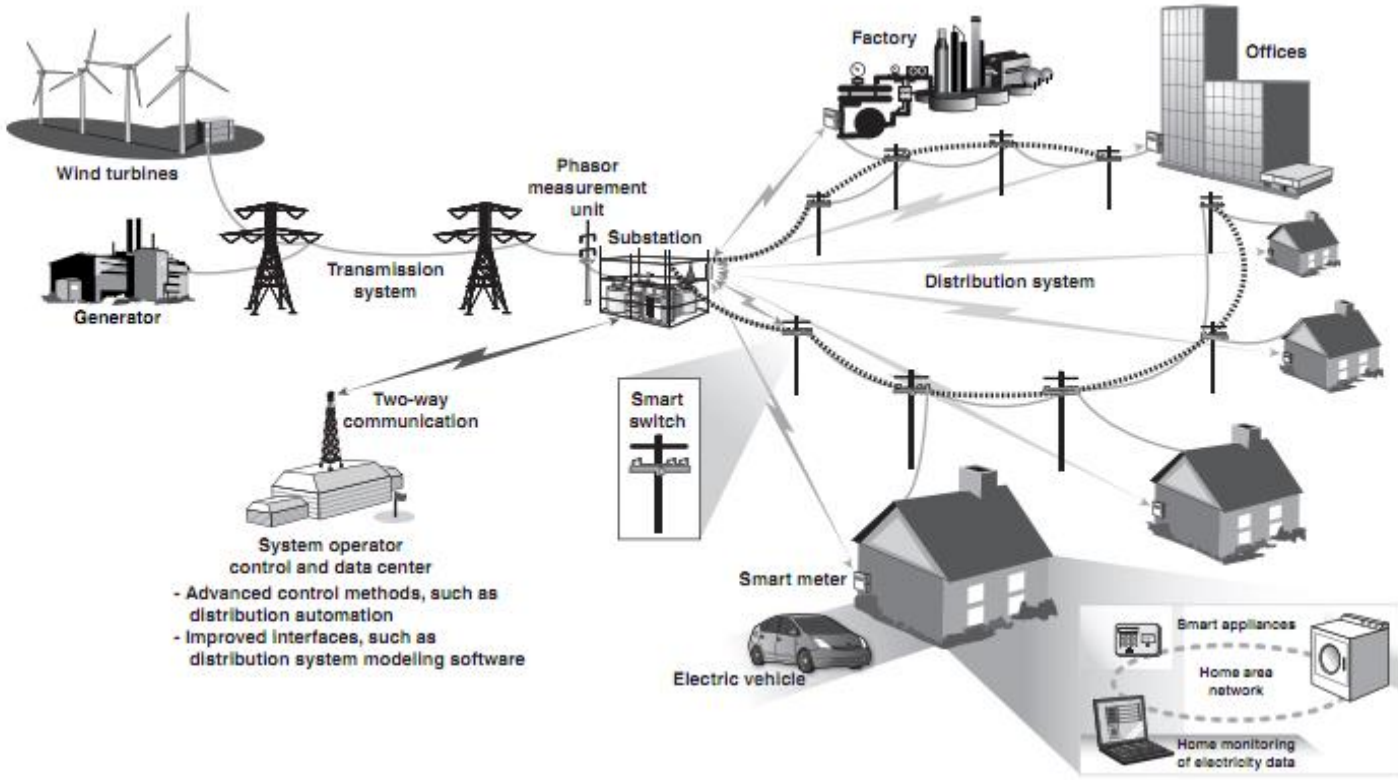
- Podívejte se na možnosti nástroje SONARQUBE
- Přijdejte se k nám na Den s průmyslovými patnery FI MU
  https://www.fi.muni.cz/for-partners/meeting-autumn2023.html.cs

VÝZNAMNOST KVALITY STOUPÁ

Střet digitalizace s kritickými infrastrukturami

Kritické infrastruktury

- **Kybernetický a fyzický** prostor se spojil v jeden
- Kdybychom zůstali digitální, moc by toho nehrozilo, ale přecházíme na **dálkové ovládání** různých kritických funkcí



Situace je komplikovaná i vlivem dalších faktorů

- **Hyperpropojený svět** a ekonomické prostředí, kaskádovité šíření problémů, nepředvídatelné dopady
- Nejistota ohledně **důvěryhodnosti připojených zařízení**
- **Vysoce distribuované prostředí**, více vstupních míst k zabezpečení, nekonzistence dat, nespolehlivé senzory, částečné poruchy.
- Nutnost zabezpečení proti **dosud neexistujícím hrozbám**

Ochrana proti dosud neexistujícím hrozbám

- Už nestačí navrhovat systémy pro **předcházení problémům**.
- Musíme předvídat **úmyslné a neúmyslné** problémy na všech úrovních.

- **Předpřipravené mechanismy pro:**
  - rozpoznání útoku/poruchy,
  - zabránit jejímu šíření,
  - zajištění základních funkcí i při útoku/poruše,
  - zotavení z útoku/poruchy,
  - forenzní analýza po útoku/poruše

Real-life story – Amazon 1p Christmas problem

Amazon

This article is more than **6 years old**

Amazon sellers hit by nightmare before Christmas as glitch cuts prices to 1p

Small businesses count cost of error in RepricerExpress software that resulted in thousands of items going for a song

Rupert Neate
@RupertNeate

Sun 14 Dec 2014 18.11 GMT

482

Most viewed


Biden-Xi virtual summit: leaders warn each other over future of Taiwan


Stanley Johnson accused of inappropriately touching Tory MP


Mysterious neurological illness haunts remote Canadian region


'The unknown is scary': why young women on social media are developing Tourette's-like tics


'It's the biggest open secret out there': the double lives of white-collar workers with

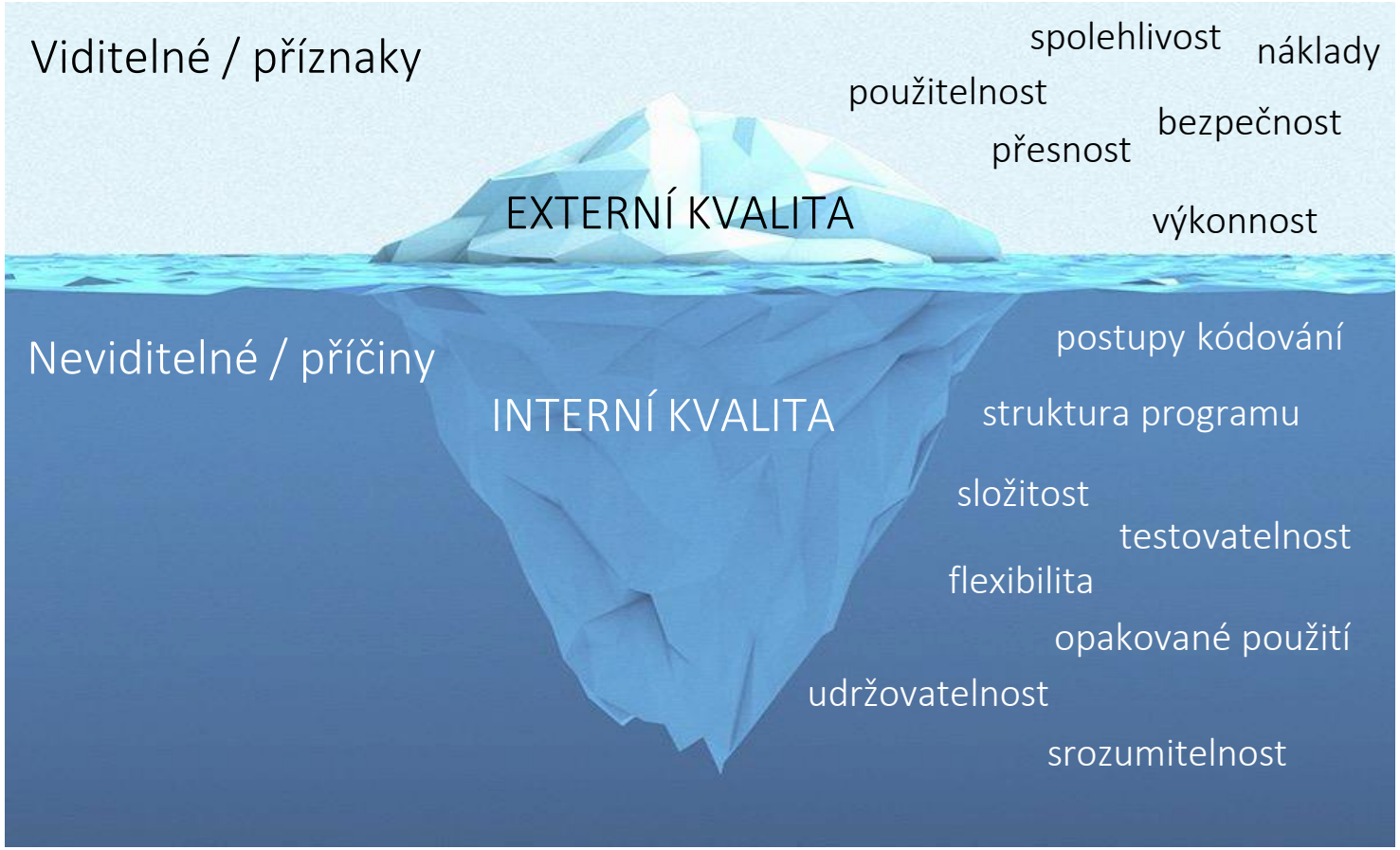
RŮZNÉ ÚHLY POHLEDU NA KVALITU SOFTWARE

MUNI FI

Funkční a nefunkční požadavky

- Funkční požadavky
  - Definice služeb, které by měl systém poskytovat, toho, jak by měl systém reagovat na určité vstupy a jak by se měl chovat v určitých situacích.
- Nefunkční požadavky
  - Vlastnosti a omezení služeb nabízených systémem, jako jsou časová, spolehlivostní a bezpečnostní omezení, omezení vývojového procesu, platformy, standardů, atd.
- Nefunkční požadavky nám pomáhají definovat
  - **Kvalitu** softwarového produktu
  - **Omezení**, tj. soulad s kontextem (organizace a legislativa)

Požadavky na produkt a kvalita SW



MUNI FI

Požadavky na kvalitu produktu

- Udržovatelnost
- Výkonnost
- Spolehlivost
- Bezpečnost
- Použitelnost
- Odolnost, robustnost
- Přenositelnost
- Přizpůsobivost
- Modularita
- Opakovaná použitelnost
- Srozumitelnost, složitost
- Testovatelnost

UDRŽOVATELNOST

MUNI FI

Udržovatelnost

- Určuje schopnost systému přizpůsobovat se změnám
- Klíčovým cílem je **předcházet technickému dluhu**
  - tj. čas, který v budoucnu strávíme refactoringem současného řešení
  - vzniká špatným návrhem nebo snahou co nejvíce urychlit proces vývoje
  - krátkodobě se může zdát výhodné rychle vytvořit funkční produkt, ten ale může být dlouhodobě neudržovatelný a vytvořit tak náklady navíc v budoucnu na nápravu

Kde je udržovatelnost důležitá?

- V systémech, které musí být funkční po mnoho let, nebo často mění nebo rozšiřují svou funkcionalitu

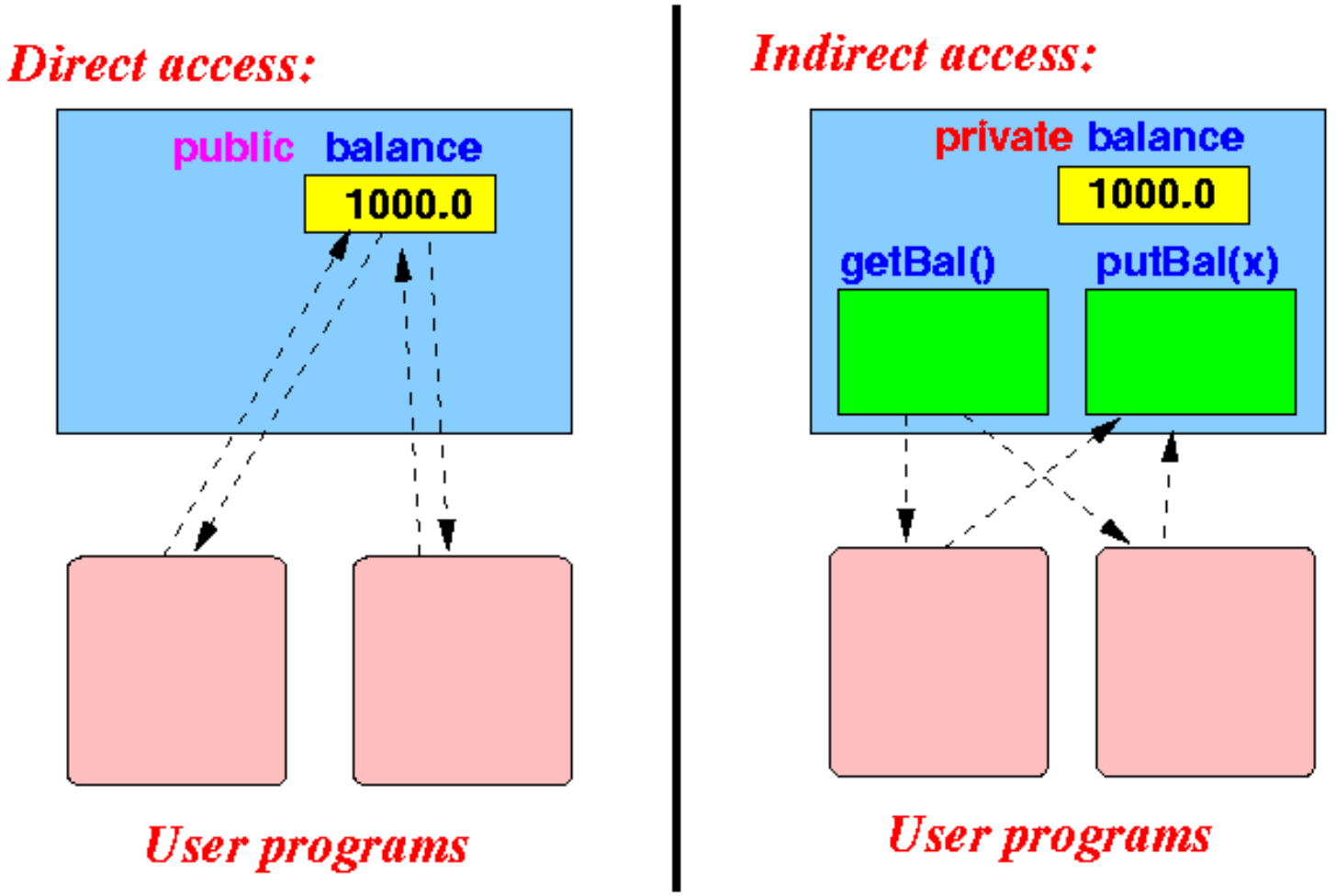
- **Například:**
  - Bankovnictví (tradiční systémy žijící dlouhou dobu)
  - Personalistika
  - Řízení dopravy
  - Sociální sítě (dynamické a rychle se měnící systémy)

Jak systém udělat udržovatelnější?

Taktiky modifikovatelnosti – předcházení tzv. ripple effects

- Lavinovým efektem změny je nutnost provést změny v modulech, kterých se změna přímo netýká.
  - Pokud je například modul A změněn tak, aby bylo možné provést určitou modifikaci, pak je modul B změněn pouze kvůli změně modulu A. Modul B musí být změněn, protože v určitém smyslu závisí na modulu A.
- **Skrýt informace.** Skrývání informací je rozklad odpovědnosti za entitu (systém nebo určitý rozklad systému) na menší části a výběr informací, které budou viditelné.

Skrývání informací



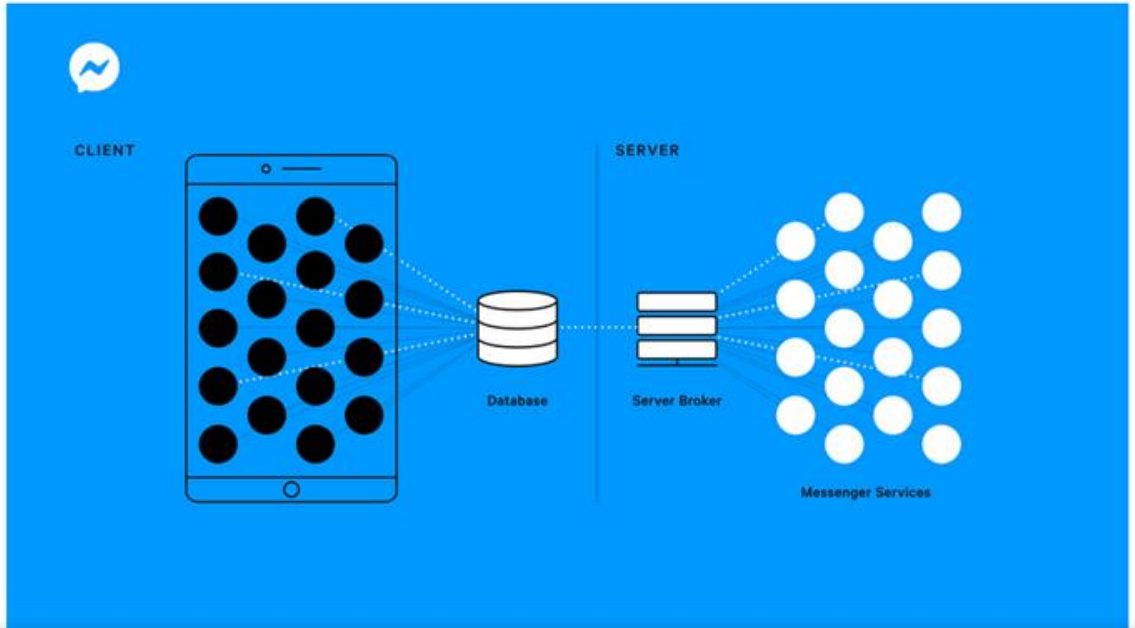
Real-life story - Rewriting the Messenger codebase

FACEBOOK Engineering

Open Source | Platforms | Infrastructure Systems | Physical Infrastructure | Video Engineering & AR/VR | Artificial Intelligence | Watch Videos

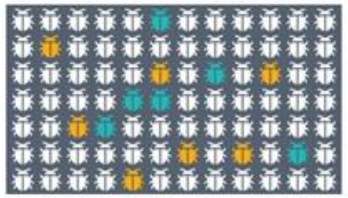
POSTED ON MARCH 2, 2020 TO ANDROID, DATA INFRASTRUCTURE, IOS

Project LightSpeed: Rewriting the Messenger codebase for a faster, smaller, and simpler messaging app



Related Posts

Dec 02, 2019
Data Transfer Project: Enabling portability of photos and videos between services



Nov 26, 2019
CCSM: Scalable statistical anomaly detection to resolve app crashes faster

Real-life story - Rewriting the Messenger codebase

- Od svého vzniku v roce 2011 byl Messenger rozšířen o mnoho nových funkcionalit, což vedlo k postupnému nárůstu počtu řádků kódu až na více než 1,7 milionu.
- Toto výrazně znesnadňovalo jakékoli úpravy aplikace, proto se v roce 2020 rozhodli návrháři přepsat ji celou od začátku a zmenšili počet řádků na 360 tisíc.
- Aby se předcházelo opětovnému nárůstu velikosti, mají nyní návrháři nových funkcí limit, který nesmí svými nápady překročit.

VÝKONNOST

MUNI FI

Výkonnost

- Výkonnost je o čase - o době odezvy na události (přerušení, zprávy, požadavky uživatelů nebo plynutí času).
  - U webového finančního systému může být odpovědí **počet transakcí, které lze zpracovat za minutu**, nebo očekávaná doba trvání jedné transakce (zadaná jako náhodná veličina).
- Velmi **citlivé na souběžné efekty** (počet uživatelů, sdílené zdroje), hardware, implementaci operačního systému (např. strategii plánovače) atd.
- Často je doprovázeno charakterizací **propustnosti a využití zdrojů**.

Kde je výkonnost důležitá?

- V systémech, kde očekáváme rychlou odezvu

- **Například:**
  - počítačové hry
  - řízení dopravy
  - biochemické simulace
  - machine learning

Jak systém udělat výkonnější?

Výkonnostní taktiky

- Zavedení souběžnosti.
  - Pokud lze požadavky zpracovávat paralelně, lze zkrátit dobu blokování. Je však nutné dobře porozumět účinkům souběhu.
- Kontrola využívání zdrojů.
  - To zahrnuje výpočetní zdroje i data. Konkrétně jde o vyvážení zátěže, řízení přístupu, plánování (prostřednictvím priority), ukládání do mezipaměti, udržování více kopií s cílem snížit spor.
- Zvýšení dostupných zdrojů.
  - Rychlejší procesory, další procesory, další paměť a rychlejší sítě mají potenciál snížit latenci.

SPOLEHLIVOST

MUNI FI

Spolehlivost (Reliability)

- Pravděpodobnost bezporuchového provozu systému po určitou dobu v daném prostředí pro daný účel.
- **Týká se**
  - Jak se zjišťuje závada/chyba/porucha systému.
  - Jak často může dojít k poruše/chybě/závadě systému.
  - Co se stane, když dojde k poruše/chybě/závadě.
- **Lze vyjádřit kvantitativně**
  - Použití pravděpodobnosti selhání na vyžádání (POFOD) v rámci jedné služby nebo provedení scénáře použití jako 1 - POFOD.

Dostupnost (Availability)

- Pravděpodobnost, že systém bude v určitém okamžiku funkční a schopen poskytovat požadované služby.
- Týká se
  - Jak dlouho by měl systém fungovat bez poruchy.
  - Jak dlouho může být systém mimo provoz.
- Lze vyjádřit kvantitativně
  - Pomocí střední doby do poruchy (MTTF) a opravy (MTTR) jako MTTF / (MTTF + MTTR).
  - Tj. dostupnost 0,999 znamená, že systém je v provozu 99,9 % času.

Nároky na spolehlivost

- Zákazníci softwaru obecně očekávají, že veškerý software bude spolehlivý. U **nekritických aplikací** však mohou být ochotni akceptovat určité selhání systému.
- Některé aplikace (**kritické systémy**) mají **velmi vysoké požadavky na spolehlivost** a k dosažení tohoto cíle mohou být použity speciální techniky softwarového inženýrství.

Kde je spolehlivost důležitá?

- V kritických infrastrukturách a projektech, kde vyjdou chyby draho

- **Například:**
  - Zdravotnictví
  - řízení letecké dopravy
  - Elektrárny
  - vesmírné mise

Jak systém udělat spolehlivější?

MUNI FI

Dosažení spolehlivosti

- Vyhýbání se poruchám
  - Vývojový proces je organizován tak, aby byly chyby v systému odhaleny a opraveny ještě před dodáním zákazníkovi.
  - Techniky verifikace a validace se používají k odhalení a odstranění chyb před nasazením
- Detekce poruch
  - Techniky detekce chyb a selhání za běhu, jako jsou akceptační testy, ping/echo, heartbeat.
- Odolnost proti chybám
  - Systém je navržen tak, aby chyby v dodaném softwaru nevedly k selhání systému.

Redundance a rozmanitost

- Redundance
  - Mějte k dispozici více než 1 verzi kritické součásti, abyste měli k dispozici zálohu.
  - Např. automatické přepnutí na záložní servery v případě poruchy.
- Diverzita
  - Stejné funkce poskytněte různými způsoby, aby nedošlo k jejich stejnému selhání.
  - Např. různé servery mohou používat různé operační systémy (např. Windows a Linux).
- Přidáním diverzity a redundance se však zvyšuje složitost, což může zvýšit pravděpodobnost chyby.
  - Někteří inženýři jsou zastánci jednoduchosti a rozsáhlého V & V, což je efektivnější cesta ke spolehlivosti softwaru.

Dohledové systémy

- Specializovaný monitorovací a řídicí systém, který je spojen s jiným systémem a který může v případě poruchy provést nouzovou akci.
  - Systém pro zastavení vlaku při průjezdu na červenou
  - Systém pro vypnutí reaktoru při příliš vysoké teplotě
- Dohledové systémy jsou redundantní, protože obsahují monitorovací a řídicí funkce, které kopírují funkce řízeného softwaru.
- Dohledové systémy by měly být různorodé a měly by používat jinou technologii než řízený software.

Real-life story - Testing software for space

THE OVERFLOW

Essays, opinions, and advice on the act of computer programming from Stack Overflow.

Find something

Latest Newsletter Podcast Company

code-for-a-living MAY 11, 2021

Testing software so it's reliable enough for space

We've talked about the engineers who write the code that operates SpaceX spaceships. Now let's talk about the people who build and maintain the tools and processes that enable the developers and ultimately, help accomplish the mission of flying astronauts to space. Stack Overflow talked with Erin Ishimoticha, an engineer in the Software Delivery Engineering…

Charles R. Martin and Ben Popper

We've talked about the engineers who write the code that operates SpaceX spaceships. Now let's talk about the people who build and maintain the tools and processes that enable the developers and ultimately, help accomplish the mission of flying astronauts to space.

Real-life story - Testing software for space

- Software používaný na lety do vesmíru má vysoké požadavky na spolehlivost.
- SpaceX používá vlastní systém pro správu verzí a změn ve svém software.
- Každá změna musí projít různými fázemi testování a být zkontrolována vícero lidmi než je umožněno ji zanést do ostrého provozu.

BEZPEČNOST

MUNI FI

Bezpečnost

- Vyjadřuje schopnost systému pracovat normálně nebo abnormálně, aniž by hrozilo nebezpečí **zranění nebo smrti člověka** a aniž by bylo poškozeno **prostředí systému**.
- Je důležité vzít v úvahu bezpečnost softwaru, protože většina zařízení, jejichž selhání je kritické, dnes obsahuje **řídicí systémy** založené na softwaru.
- Požadavky na bezpečnost jsou často požadavky výlučné, tj. **vylučují nežádoucí situace**, spíše než aby specifikovaly požadované služby systému. Ty vytvářejí požadavky na funkční bezpečnost.

Zabezpečení

- Vlastnost systému, která odráží schopnost systému chránit se před náhodným nebo úmyslným vnějším útokem.
- Chrání systém proti:
  - **Confidentiality** – Ohrožení důvěrnosti systému a jeho údajů

    Může zpřístupnit informace osobám nebo programům, které nemají oprávnění k přístupu k těmto informacím.
  - **Integrity** – Ohrožení integrity systému a jeho dat

    Může dojít k poškození nebo zničení softwaru nebo jeho dat.
  - **Availability** – Ohrožení dostupnosti systému a jeho dat

    Může omezit přístup k systému a datům pro oprávněné uživatele.

Kde je bezpečnost důležitá?

- V systémech pracujících s citlivými/cennými daty

- **Například:**
  - Zdravotnictví
  - Bankovnictví
  - Elektrárny
  - Vojenství
  - Státní správa

Jak systém udělat bezpečnější?

Návrh pro bezpečnost

- Při návrhu pro bezpečnost je třeba zvážit dva základní aspekty
  - **Ochrana –** Jak by měl být systém organizován, aby bylo možné chránit aktiva před vnějším útokem?
  - **Distribuce –** Jak by měly být rozděleny prostředky, aby byly minimalizovány následky úspěšného útoku?

- **Potenciálně protichůdné**
  - Pokud je majetek rozdělen, je jeho ochrana nákladnější. Pokud jsou aktiva chráněna, mohou být ohroženy požadavky na použitelnost a výkon.

Bezpečnostní pokyny

Bezpečnostní taktiky
Zakládat zabezpečení na explicitní bezpečnostní politice
Vyhněte se jedinému bodu selhání
Bezpečné selhání
Vyvážení bezpečnosti a použitelnosti
Zaznamenávání akcí uživatele
Využití redundance a diverzity ke snížení rizika
Rozdělení majetku
Návrh pro obnovitelnost
Ověření všech vstupů

Bezpečnostní doporučení 1-3

- Zakládat zabezpečení na explicitní bezpečnostní politice
  - Definujte bezpečnostní politiku organizace, která stanoví základní bezpečnostní požadavky, jež by měly platit pro všechny systémy organizace.
- Vyhněte se jedinému bodu selhání
  - Zajistěte, aby k selhání zabezpečení mohlo dojít pouze v případě, že dojde k více než jednomu selhání bezpečnostních postupů. Mějte například ověřování založené na heslech a otázkách.
- Bezpečné selhání
  - Pokud systémy z jakéhokoli důvodu selžou, zajistěte, aby k citlivým informacím neměli přístup neoprávnění uživatelé, i když nejsou k dispozici běžné bezpečnostní postupy.

Bezpečnostní doporučení 4-6

- Vyvážení bezpečnosti a použitelnosti
  - Snažte se vyhnout bezpečnostním postupům, které ztěžují používání systému. Někdy je nutné přijmout slabší zabezpečení, aby byl systém použitelnější.
- Zaznamenávání akcí uživatele
  - Udržujte protokol uživatelských akcí, který lze analyzovat a zjistit, kdo co udělal. Pokud uživatelé o takovém záznamu vědí, je méně pravděpodobné, že se budou chovat nezodpovědně.
- Využití redundance a diverzity ke snížení rizika
  - Uchovávejte více kopií dat a používejte různorodou infrastrukturu, aby zranitelnost infrastruktury nemohla být jediným bodem selhání.

Bezpečnostní doporučení 7-9

- Rozdělení přístupu
  - Uspořádejte systém tak, aby aktiva byla v oddělených oblastech a uživatelé měli přístup pouze k informacím, které potřebují, a ne ke všem systémovým informacím.
- Návrh pro obnovitelnost
  - Navrhněte systém tak, aby zjednodušil obnovitelnost po úspěšném útoku.
- Ověření všech vstupů
  - Zkontrolujte, zda jsou všechny vstupy v rozsahu, aby neočekávané vstupy nemohly způsobit problémy.

Nouzový režim (Survivability)

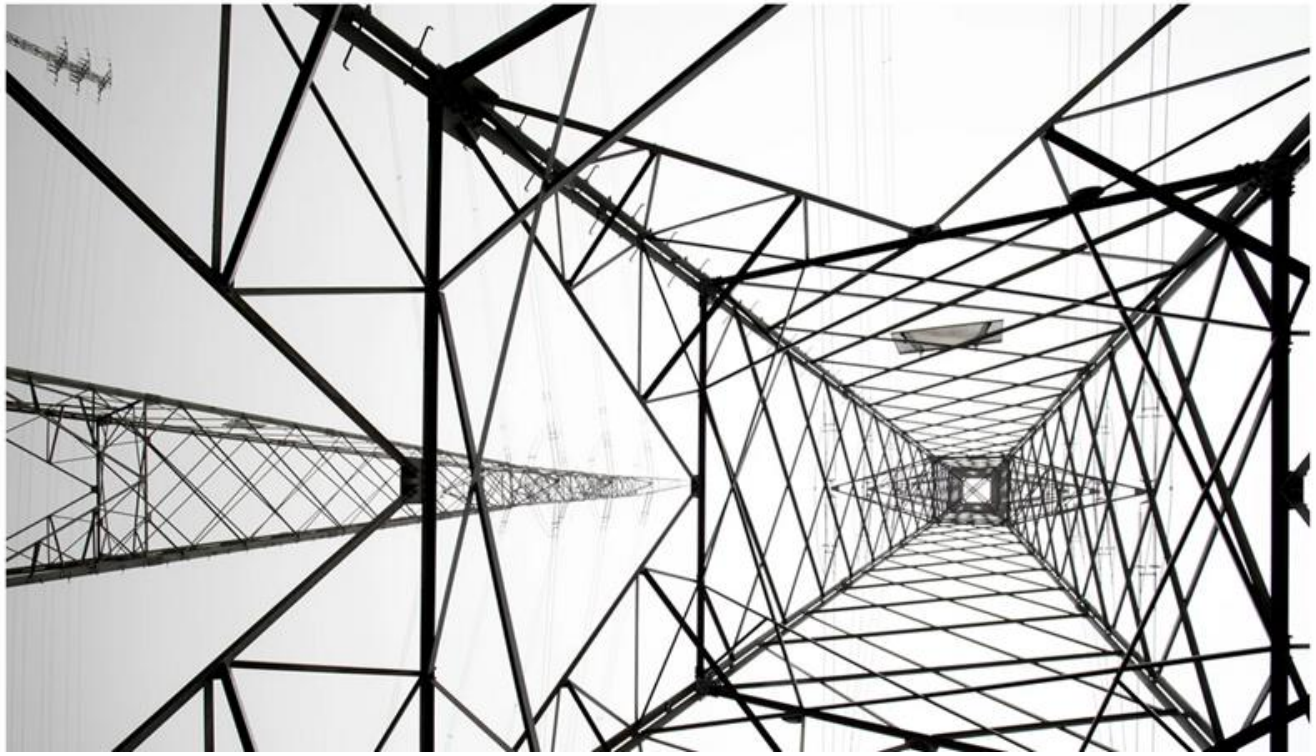
- **Nouzový režim** = schopnost systému poskytovat základní služby, i když je napaden nebo poté, co byla jeho část poškozena.
- Odolnost
  - Předcházení problémům zabudováním schopností systému odolávat útokům.
- Detekce
  - Odhalování problémů díky funkcím systému, které umožní odhalit útoky a selhání.
  - Důležitá role **monitorování a zpracování událostí**
- Zotavení
  - Zvládnutí problémů zabudováním schopností poskytovat služby i po napadení systému

Real-life story - Hack of Ukraine's Power Grid

KIM ZETTER SECURITY 03.03.2016 07:00 AM

Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid

The hack on Ukraine's power grid was a first-of-its-kind attack that sets an ominous precedent for the security of power grids everywhere.



Real-life story - Hack of Ukraine's Power Grid

- Příklad projektu, který selhal na všech úrovních.
- Nepoznali, že po dobu 6 měsíců byly sbírány přihlašovací údaje uživatelů.
- Když viděli, jak se jejich kurzory pohybují po obrazovce a vypínají rozvodny, nedokázali to zastavit.
- Nebyli schopni zajistit bezpečnost, dokonce i jejich vlastní záložní energie byla vypnuta a firmware smazán.
- Trvalo měsíce, než se z toho systém vzpamatoval.

POUŽITELNOST

MUNI FI

Použitelnost

- Použitelnost se zabývá tím, jak **snadno** může uživatel **provést požadovaný úkol** a jakou podporu uživateli systém poskytuje.
- Lze ji rozdělit do následujících oblastí:
  - Funkce výukového systému.
  - Efektivní používání systému.
  - Přizpůsobení systému potřebám uživatelů.
  - Zvyšování důvěry a spokojenosti.
- Vždy dodržujte pokyny pro lidské rozhraní (HIG), pokud jsou k dispozici (HIG pro Windows, HIG pro Mac OS a další).

Jak systém udělat použitelnější?

MUNI FI

Taktiky použitelnosti – viz uživatelská přívětivost

- Udržujte model úkolu.
  - Model úlohy slouží k určení kontextu, aby systém měl představu o tom, o co se uživatel pokouší, a mohl mu poskytnout různé druhy pomoci.
- Udržujte model uživatele.
  - Model určuje znalosti uživatele o systému, chování uživatele z hlediska očekávané doby odezvy a další aspekty specifické pro uživatele nebo třídu uživatelů.
- Udržujte model systému.
  - Model určuje očekávané chování systému, aby bylo možné poskytnout uživateli vhodnou zpětnou vazbu.

KONFLIKTY KVALITY

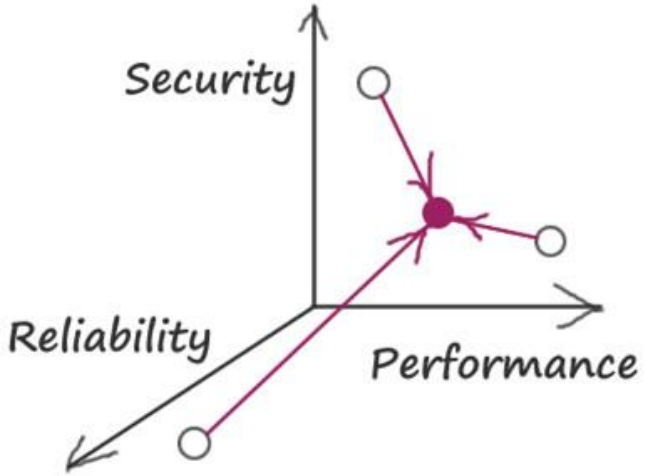
MUNI FI

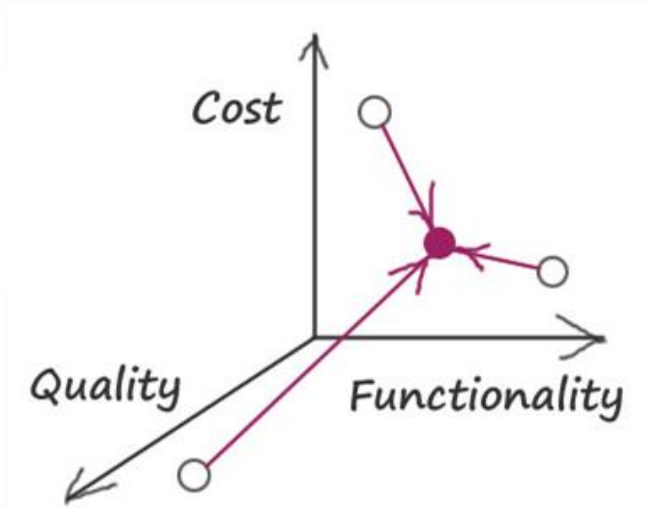
Konflikty kvality

- V rámci komplexních systémů nelze nikdy dosáhnout atributů kvality izolovaně.
  - Úspěch každého z nich bude mít vliv, někdy pozitivní a někdy negativní, na úspěchy ostatních.
- Například téměř každý atribut kvality negativně ovlivňuje výkonnost.
  - Spolehlivost. Redundance spolu s hlasovacím schématem zpožďuje odezvu systému.
  - Přenositelnost. Hlavní technikou pro dosažení přenositelného softwaru je izolace závislostí systému, která do jeho provádění vnáší režii.

Konflikty kvality

- Žádný systém nemůže být optimalizován pro všechny tyto vlastnosti.
- Plán kvality by proto měl definovat nejdůležitější atributy kvality vyvíjeného softwaru.





ROLE S VLIVEM NA KVALITU

Role s vlivem na kvalitu

- Udržovatelnost – softwarový vývojář, architekt, analytik
- Výkonnost – softwarový vývojář, performance engineer
- Spolehlivost – softwarový vývojář, tester, QA engineer
- Bezpečnost – softwarový vývojář, penetration tester
- Použitelnost – UI/UX designer

CO NÁS ČEKÁ PŘÍŠTĚ

MUNI FI

9. Algoritmické myšlení II

- MIT App Invertor
- Interaktivní demo na vytvoření chatovací aplikace

Domácí práce a příprava na tuto přednášku

- Prosím přineste si vlastní notebooky a na své chytré telefony (pokud máte) si nainstalujte MIT AI2 Companion https://appinventor.mit.edu/explore/ai2/setup-device-wifi