Chyby software

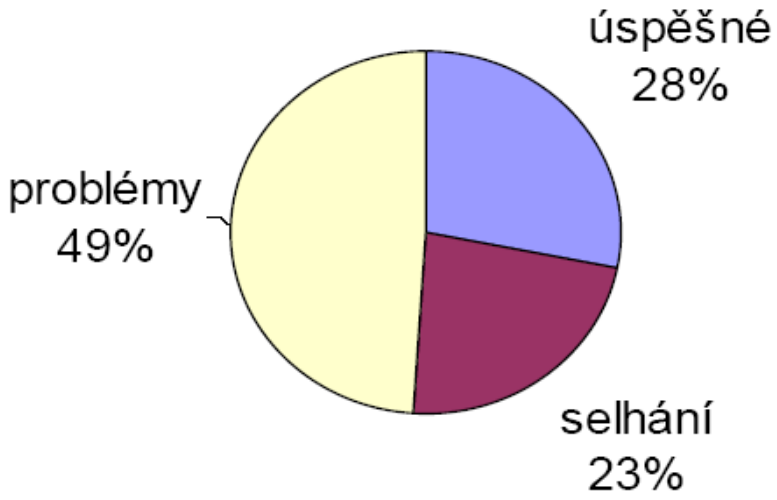
Výsledek projektu

- Úspěšný: Projekt je dokončen včas, bez překročení rozpočtu, se všemi specifikovanými rysy a funkcemi.
- S výhradami: Projekt je dokončen a funkční, ale překročil rozpočet, opožděný, méně rysů a funkcí, než bylo původně specifikováno.
- Neúspěšný: Projekt je zastaven před dokončením, není implementován, nebo vyřazen po instalaci.

Zdroj: Johnson 2001 - Standish Report

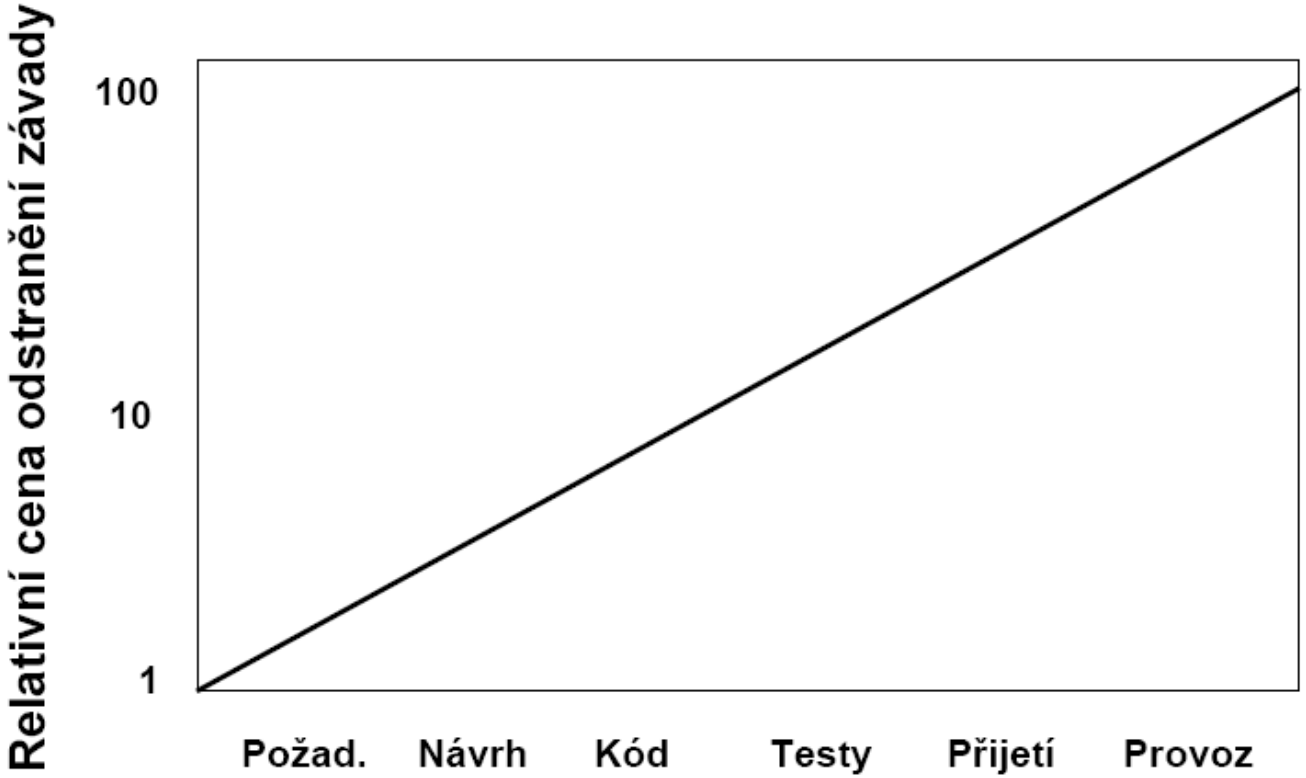
Neúspěšné projekty

- USA: “Přestaňme vyhazovat $78 bilionů ročně”
- UK: Vnitřní správa: 12 z 18 velkých IT projektů neuspělo: pasy, zdravotnictví, ...



USA: 2000 Standish Report

Relativní cena odstranění závady



Zdroj: Barry W. Boehm, 1981, COCOMO

Prevence proti neúspěchu projektu

- ZAPOJENÍ vrcholového řízení a koncových uživatelů
- POUŽITÍ efektivního řízení projektu se spoluúčastí a zapojením vrcholového řízení na přezkoumáních
- POUŽITÍ efektivního řízení požadavků
- POUŽITÍ inkrementálního vývoje
- UČINĚNÍ "všech smysluplných kroků" při inženýrských aktivitách, t.j. dokumentace, měření, plánování, sledování, řízení kvality...

⇨ Menší pravděpodobnost (totálního) neúspěchu projektu

Spolehlivost SW

- Porucha - neschopnost systému nebo systémové komponenty provádět požadovanou funkci ve specifikovaných hranicích. Porucha může nastat, když se narazí na chybu, jejímž výsledkem je ztráta očekávané uživatelské služby.

- Četnost poruch
    1. Poměr počtu chyb dané kategorie nebo významu k časovému intervalu; např. poruchy za měsíc. (jiný název: intenzita poruch)
    2. Poměr počtu poruch k dané jednotce měření; např. poruchy za jednotku času, poruchy pro daný počet transakcí, poruchy pro daný počet běhů programu.

Spolehlivost SW - Chyby

- Fault - chyba (defekt)
    1. Chyba v kódu, která může být příčinou jednoho nebo více selhání.
    2. Náhodná podmínka, která způsobuje, že funkční jednotka selhává při plnění požadované funkce. (synonymum: bug)
- Error - chyba (omyl) - nesprávná nebo chybějící akce uživatele, která zapříčiní chybu (defekt) v programu.

Spolehlivost SW - chyby

- Četnost chyb na začátku systémových testů je v rozsahu 1 až 10 chyb/KSLOC, s průměrem 6 chyb/KSLOC.

- KSLOC - počet předaných proveditelných řádek zdrojového kódu, bez znovupoužitého kódu, deklarací dat, komentářů apod. Očekávaný počet chyb odstraněných při opravě jednoho selhání: 0.955 chyb

Hustoty chyb

Aplikace	Systémy	KSLOC	Průměr chyby/KSLOC	Standardní odchylka
Vzdušné	7	541	12.8	9.4
Strategické	21	1,794	9.2	14.0
Taktické	5	88	7.8	6.1
Řízení procesů	2	140	1.8	0.3
Výroba	12	2,575	8.5	9.5
Vývojové	4	97	12.3	9.3
Celkem/Průměr	**51**	**5,236**	**9.4**	**11.0**

MS: 5 až 15 chyb/KSLOC

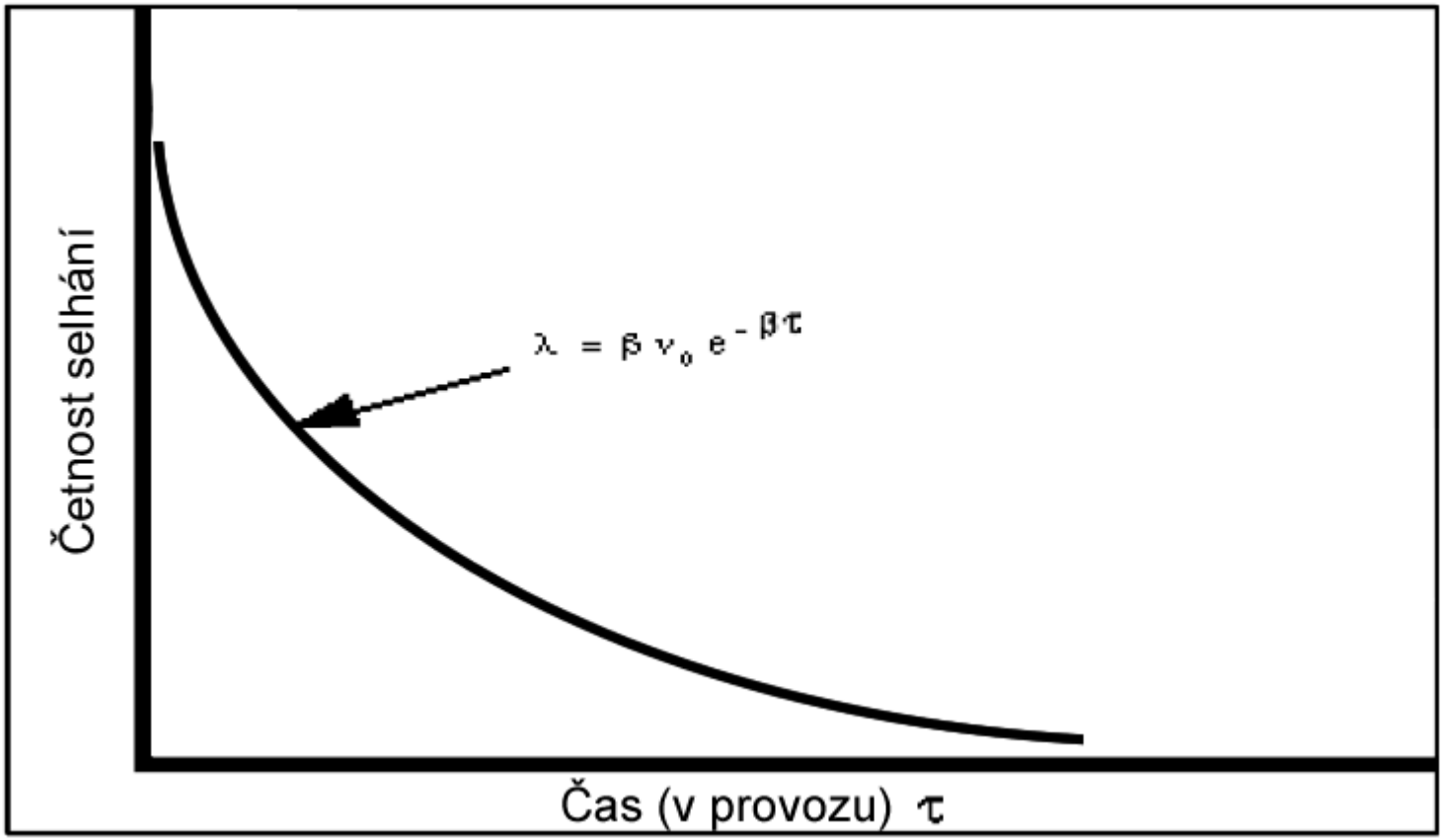
Bruce Schneier, Counterpane Internet Security Inc. , San Jose, uvedl, že Windows 2000:

- obsahuje 40 - 60 milionů řádek kódu
- 5 - 15 chyb/tisíc řádek kódu.

Dále uvedl, že ačkoliv Microsoft vyvinul značné úsilí odstranit chyby před uvedením produktu na trh, pro odstranění chyb potřebuje společnost najmout 2x více lidí, než bylo bylo zapojeno do návrhu.

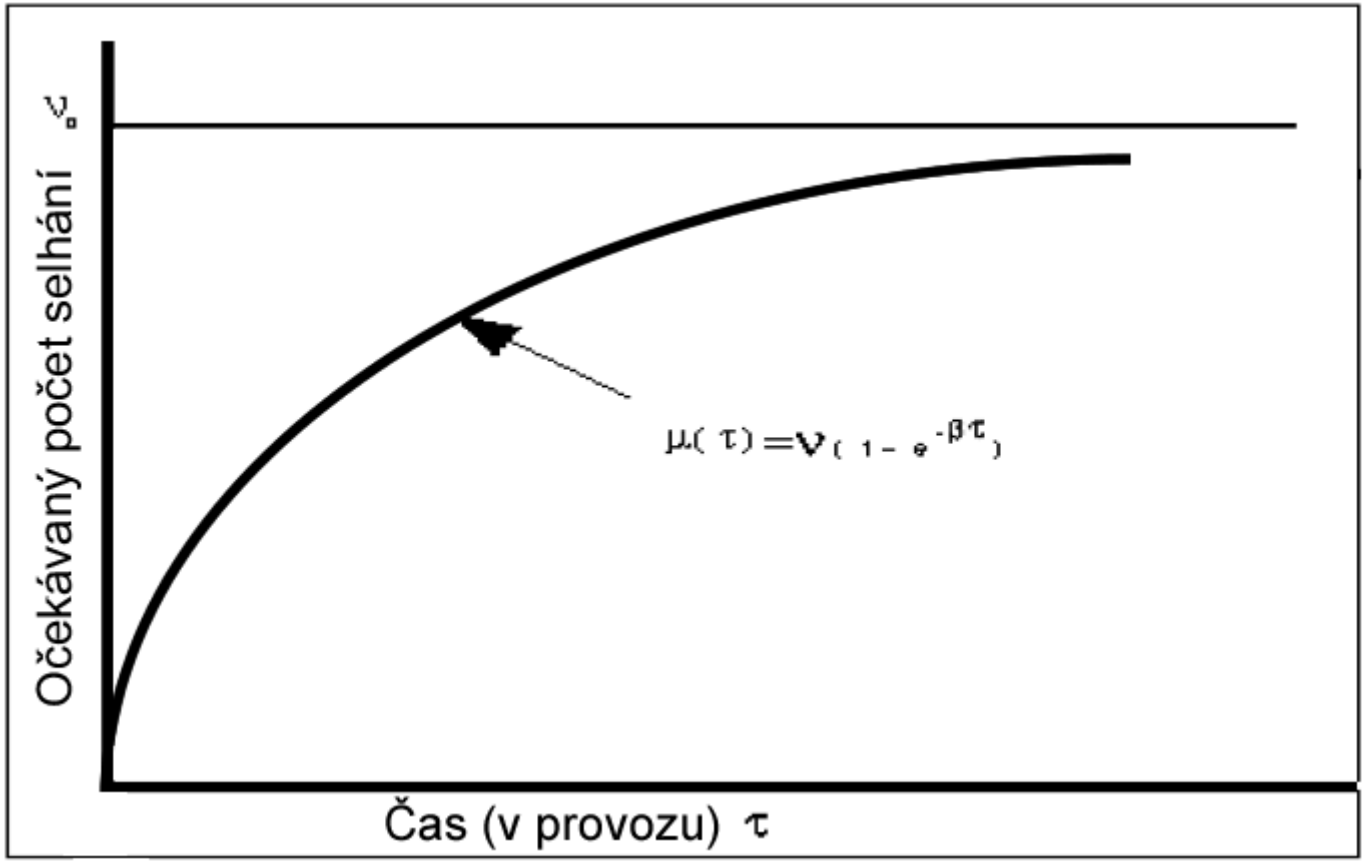
„Složitost je nepřítelem bezpečnosti. Jak [Windows 2000] vzrůstá složitost, vidíme stále více chyb."

Četnost selhání v čase



Musa

Očekávaná neodhalená selhání



Musa

Některé druhy chyb

- algoritmická chyba
- chyba syntaxe
- chyba výpočtu a přesnosti
- chyba dokumentace
- chyba stresu nebo přetížení
- chyba kapacity nebo meze
- časová nebo součinnostní chyba
- chyba propustnosti nebo výkonu
- chyba zotavení
- chyba HW a systémového SW
- chyba nedodržení standardů a procedur

IBM ortogonální klasifikace defektů (ODC)

- Funkce - chyba ovlivňující schopnosti, rozhraní uživatelů, rozhraní výrobku, rozhraní s HW architekturou nebo globální datovou strukturou.
- Rozhraní - chyba při interakci s ostatními komponentami nebo ovladači přes volání, makra, řídící bloky nebo seznamy parametrů.
- Ověřování - chyba v logice programu, která selže při validaci dat a hodnot před tím, než jsou použity.
- Přiřazení - chyba při inicializaci datové struktury nebo bloku kódu.
- Časování/serializace - chyba, která zahrnuje časování sdílených a RT prostředků.
- Sestavení/balení/spojování - chyba související s problémy s repozitory projektu, změnami vedení, nebo správou verzí.
- Dokumentace - chyba, která ovlivňuje publikace a návody pro údržbu.
- Algoritmus - chyba, která se týká efektivity nebo správnosti algoritmu nebo datové struktury, ne však jejich návrhu.

OCD atributy typů defektů

- Programátor, který řeší defekt, obvykle zvolí typ defektu. Volba typu defektu vyplývá z eventuální opravy. Typy jsou jednoduché, tak aby byly jasné programátorovi a omezily prostor pro případná nedorozumění.
- Vždy se rozliší mezi tím, že něco *chybí* nebo je něco *nesprávné*.
- *Funkční* chyba je ta, která ovlivňuje významně schopnosti, rozhraní koncových uživatelů, rozhraní výrobku, rozhraní s HW nebo globální datové struktury a bude vyžadovat formální změnu návrhu.
- Chyba *přiřazení* indikuje několik řádek kódu, jako je inicializace řídících bloků nebo datových struktur.
- *Rozhraní* odpovídá chybám při interakci s jinými komponentami, moduly nebo ovladači zařízení pomocí maker, příkazů volání, řídících bloků nebo seznamu parametrů.

Chillarege et al

OCD atributy typů defektů

- *Ověřování* se týká logiky programu, která neuspěla při řádné validaci dat a hodnot před jejich použitím.
- Chyby *časování/serializace* jsou ty,které lze opravit zlepšeným řízením sdílených a RT prostředků.
- *Sestavení/balení/spojování* popisuje chyby, které se objevují díky chybám ve správě knihoven, řízení změn nebo ve správě verzí.
- *Dokumentační* chyby mohou ovlivnit jak publikace, tak údržbovou dokumentaci.
- *Algoritmické* chyby zahrnují problémy efektivity nebo správnosti, které mají vliv na úlohy a mohou být odstraněny pomocí (re)implementace algoritmu nebo lokální datové struktury bez potřeby požadavku na změnu v návrhu.

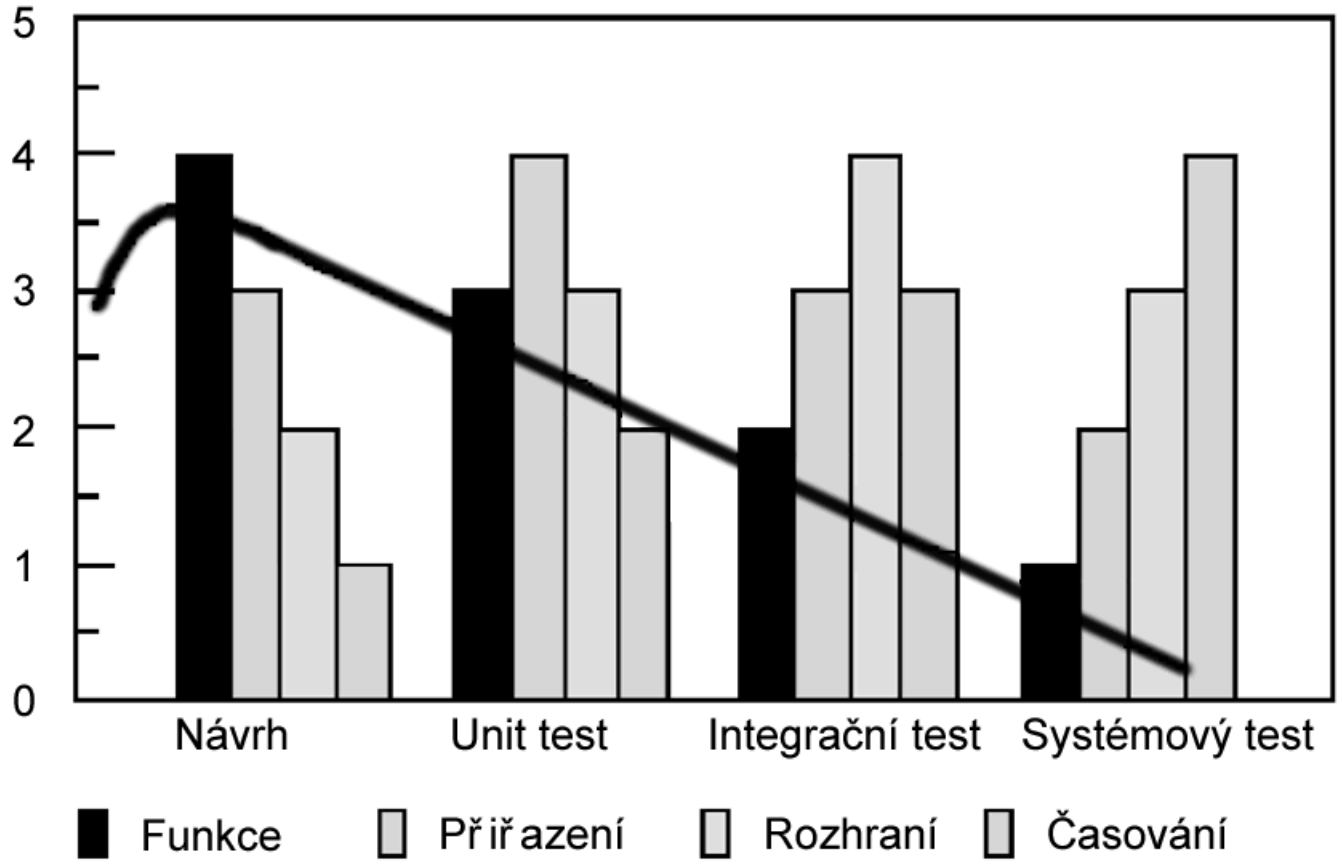
Chillarege et al

Typ defektu a asociace s etapou

Typ defektu	Vývojová etapa
• funkce	• návrh
• rozhraní	• návrh na nízké úrovni
• ověřování	• návrh na nízké úrovni nebo kód
• přiřazení	• kód
• časování/serializace	• návrh na nízké úrovni
• sestavení/balení/spojování	• knihovní nástroje
• dokumentace	• publikace
• algoritmus	• návrh na nízké úrovni

Chillarege et al

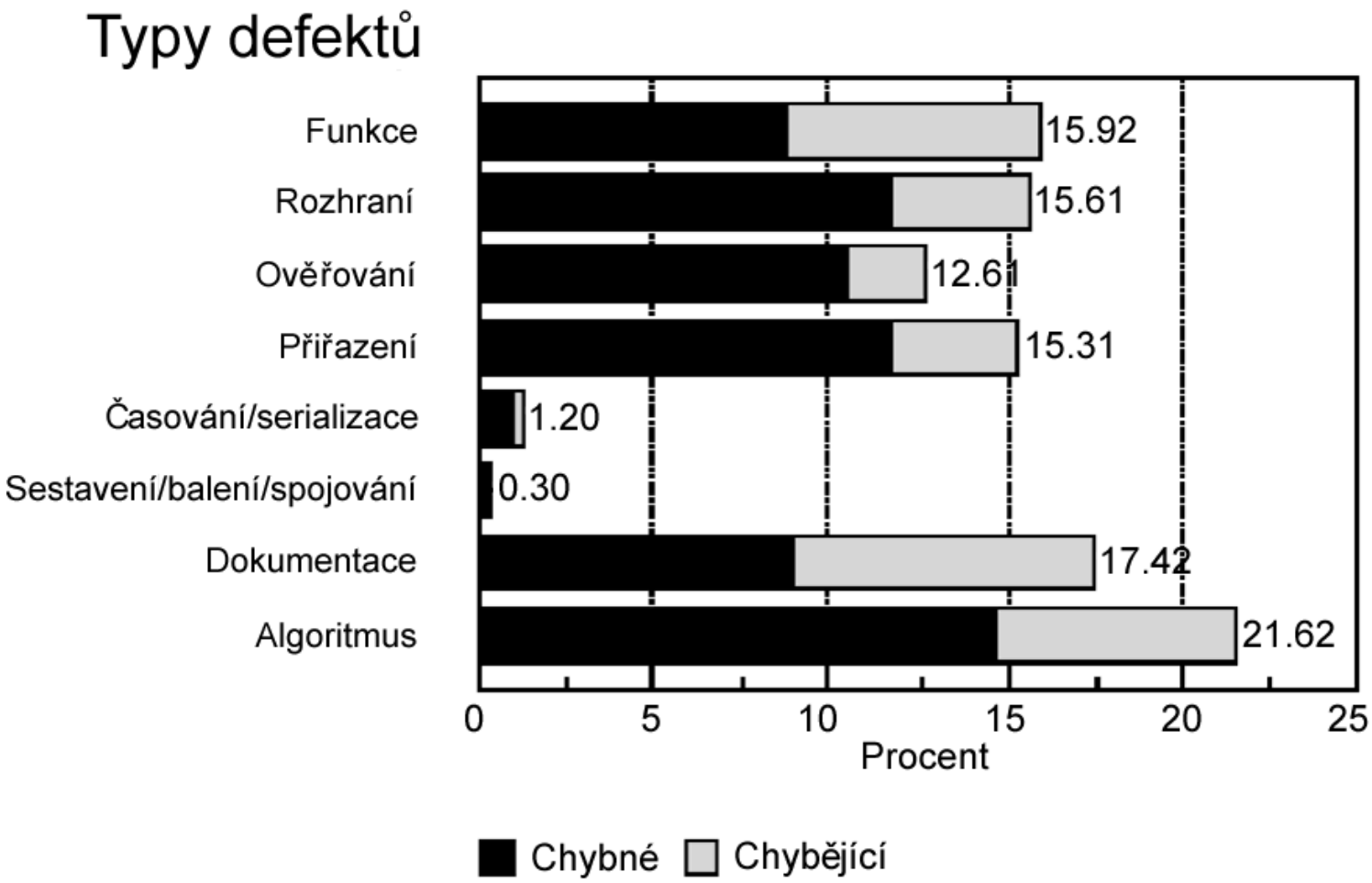
Funkční selhání podle etapy



Celkové defekty a defekty typu *Funkce*

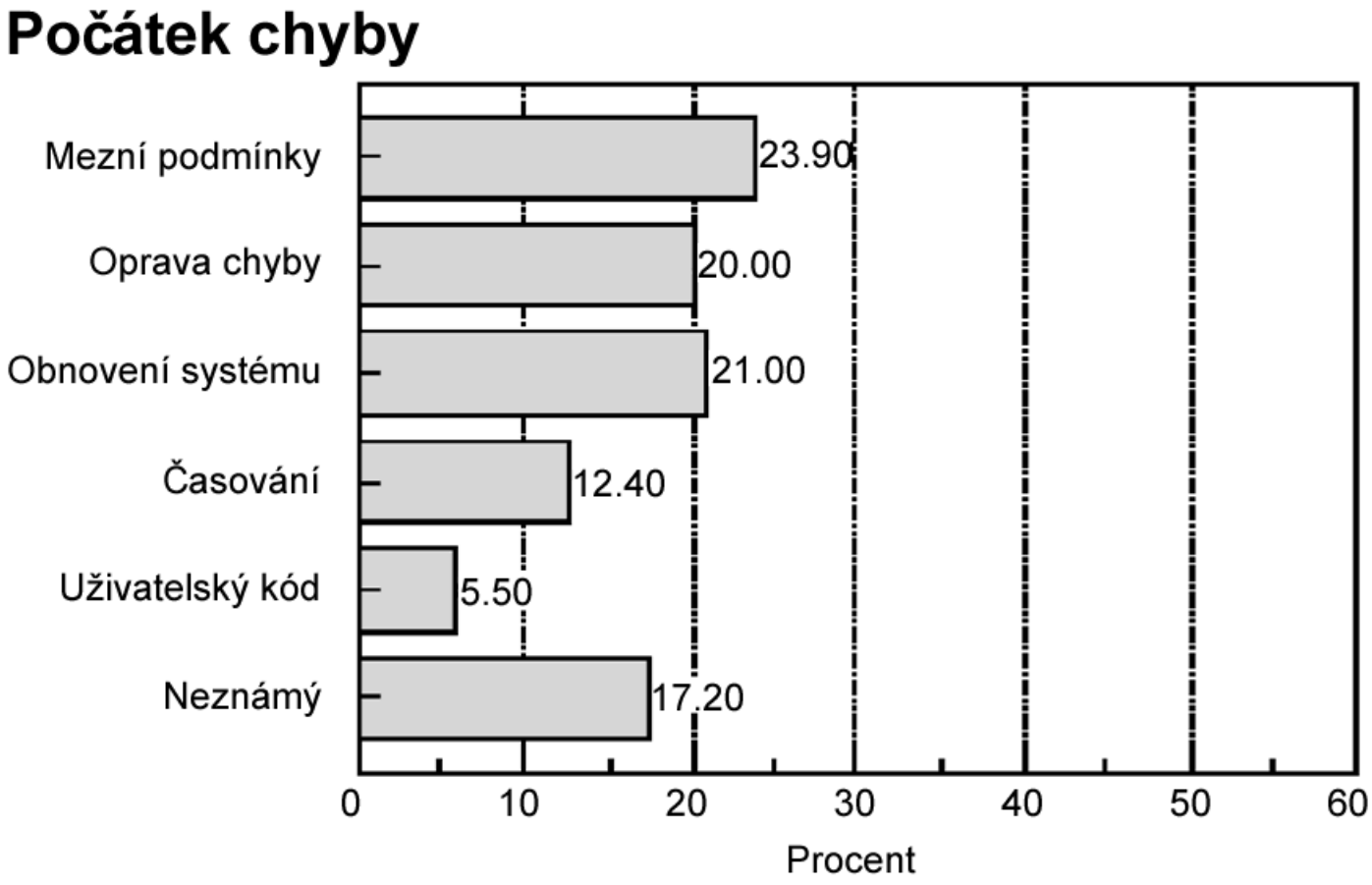
Chillarege et al

Rozložení druhů chyb při testování



Chillarege et al

Rozložení počátku (spouště) chyby



(systémy DB2 a IMS)

MVS

Chillarege et al

Odkazy

- *Software Engineering Baselines*
  http://www.dacs.dtic.mil/techs/baselines/reliability.html
- Musa J.D., Iannino A., Okumoto K., *Software Reliability*, McGraw Hill, 1987
- Musa J., *Software Reliability Engineering*, McGraw Hill, 1999
- John D. Musa, AT&T Bell Laboratories and James Widmaier,
- National Security Agency, *Software-Reliability-Engineered Testing,*
  http://www.stsc.hill.af.mil/crosstalk/1996/jun/Reliabil.asp
- S.L. Pleeger, *Software Engineering*, 2ed – Sections 8.1-8.3
- Ram Chillarege et al, *Orthogonal Defect Classification - A Concept for In-Process Measurements,* IEEE Transactions on Software Engineering, Vol 18, No. 11, Nov 1992. Copyright 1992 IEEE,
  http://www.chillarege.com/odc/articles/odcconcept/odc.html,
  http://www.chillarege.com/odc/