

Zavedení

Jak instalovat a zavádět

Na koho se obrátit

Pozor

- Zavedení je společná práce
 - Manažerů dodavatele – zajistí dodávku systému
 - Manažerů odběratele – zajistí lidi, prostředky, organizaci na místě
 - Expertů resp uživatelů obou stran

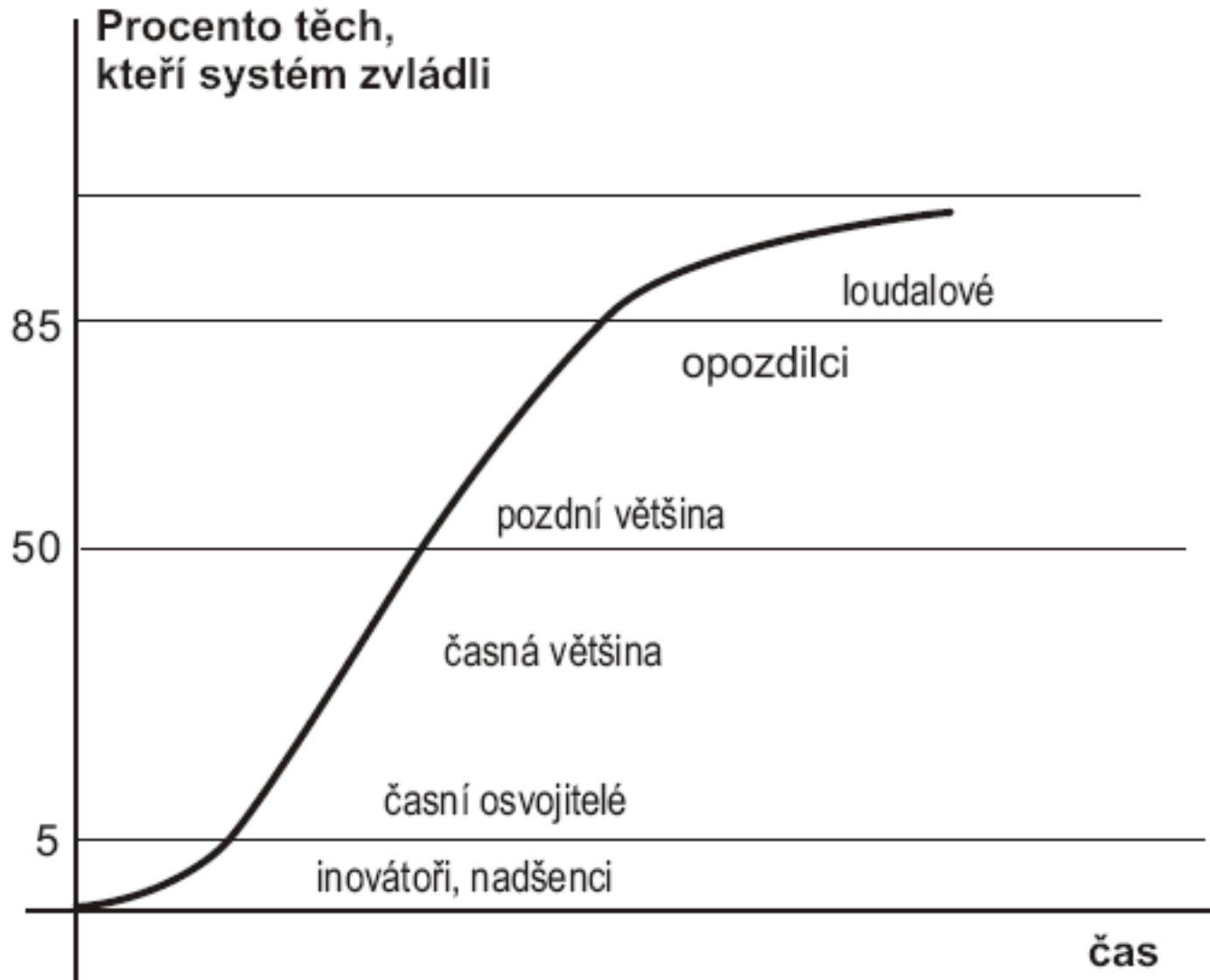
Zavedení informačního systému

- Předání (instalace, předávací testy, zkušební provoz)
- Školení pracovníků
 - Spoluúčast tvůrců (nebývají ale pedagogicky zdatní, nevyužívá se jejich odbornost)
 - Vhodní jsou dobří lektoři, dokonce specializované firmy
- Plánování přechodu
 - Konverze dat Postup překlopení
 - Vhodné je inkrementální zavádění, lze-li. Podmínkou je vhodná architektura.
 - Je vhodné stanovit kriteria úspěchu zavádění

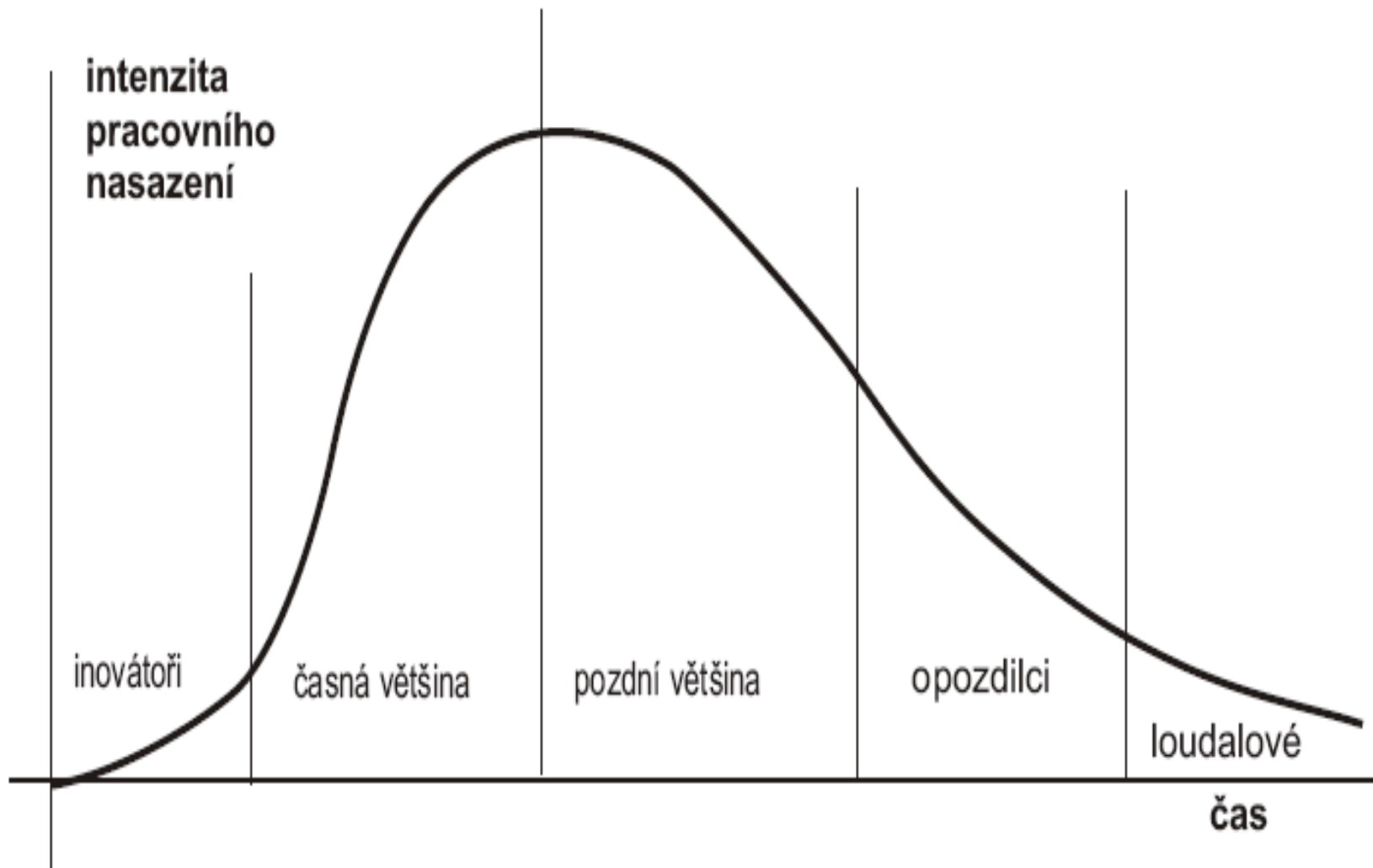
Dokumenty při zavádění

- Vize systému a specifikace požadavků
- Návrh a zdrojové texty (nedělá-li dodavatel údržbu)
- Dokumenty o testech, případně deník projektu
- Manuály
- Dohoda o zkušební době a procedurách odstraňování chyb
- Záruky a rizika

Křivka učení a typy uživatelů



Křivka učení a typy uživatelů



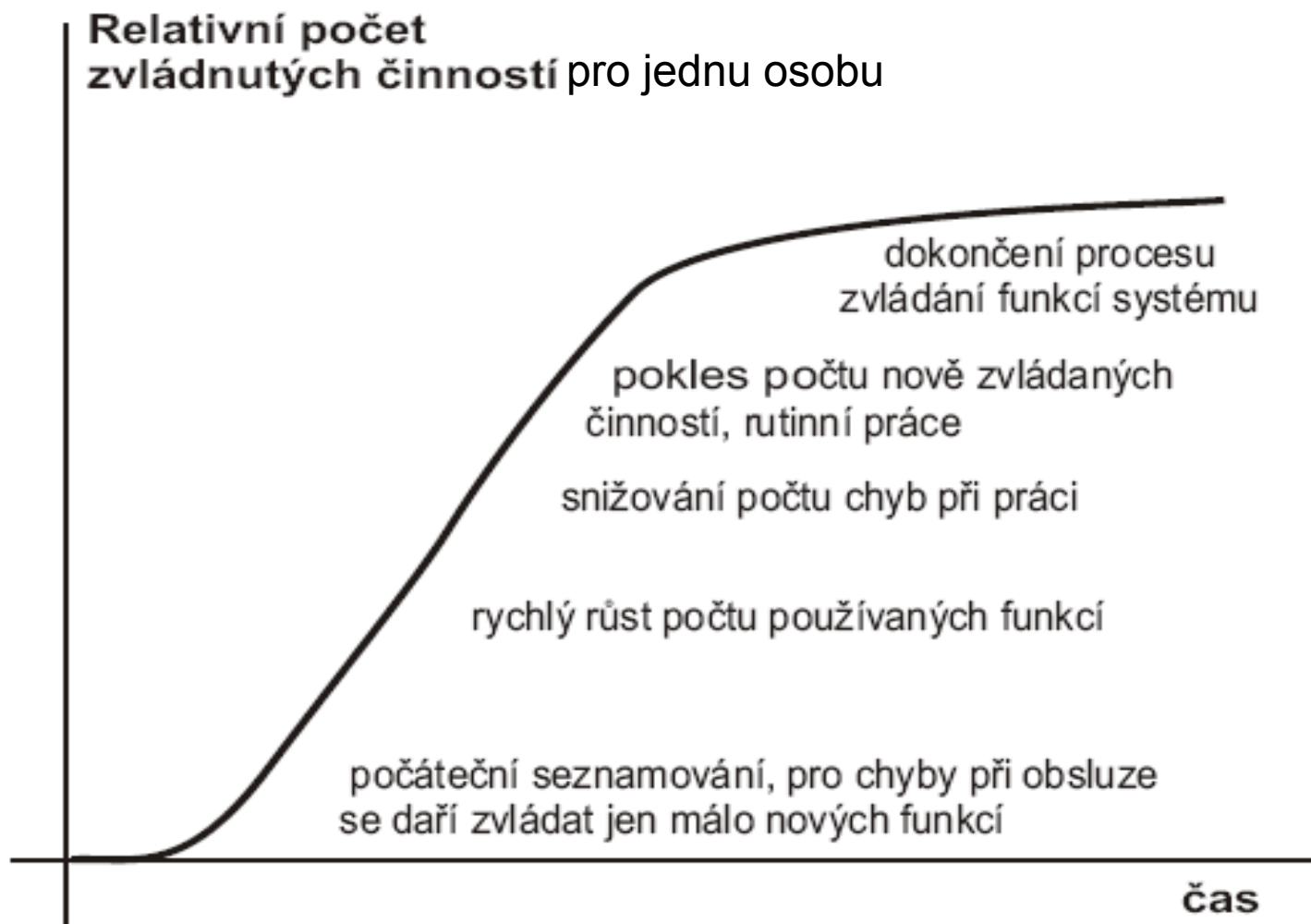
Koho získat jako spojence

- Inovátoři jsou motivováni novinkami, nikoliv zlepšováním své práce,
 - Nemívají prestiž mezi spolupracovníky, nerozumí jejich potřebám
 - Brzy ztrácí zájem
- Časní osvojitelé chtějí zlepšovat svou práci a noviny při tom vítají, mívají vysokou reputaci u uživatelů, to jsou ti praví spojenci
- Časná většina inovace spíše vítá, ale chce mít svůj klid

Křivka učení

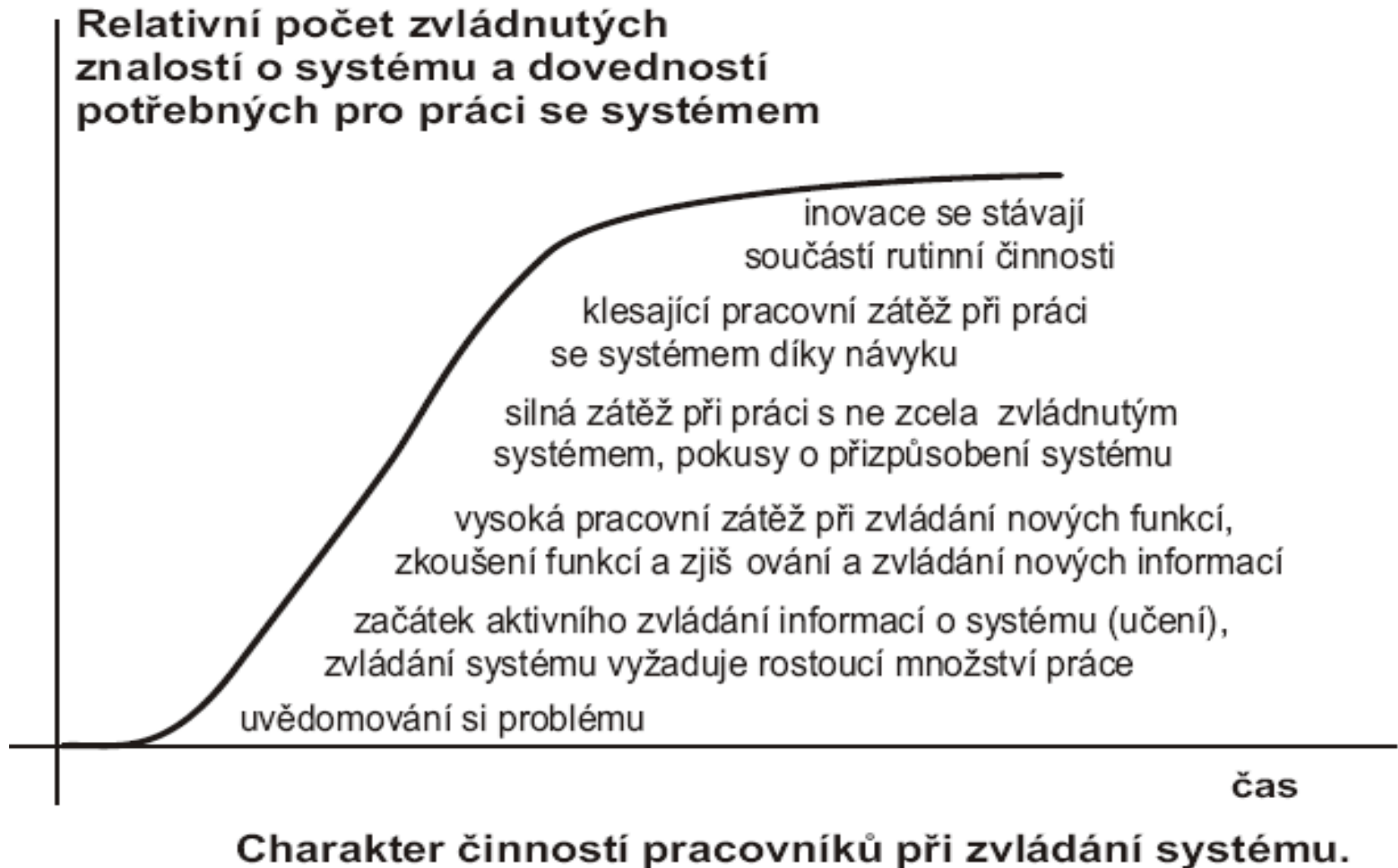
- Nemá být příliš strmá (intensita učení příliš vysoká) ani příliš plochá
- Je nutné zvládnání chápat jako učení, je to tedy specifický úkol a je proto třeba zajistit
 - Kvalitní vyučující
 - Čas, pomůcky a prostředí
 - Hodnocení pokroku („známkování“)
- Často se to podceňuje

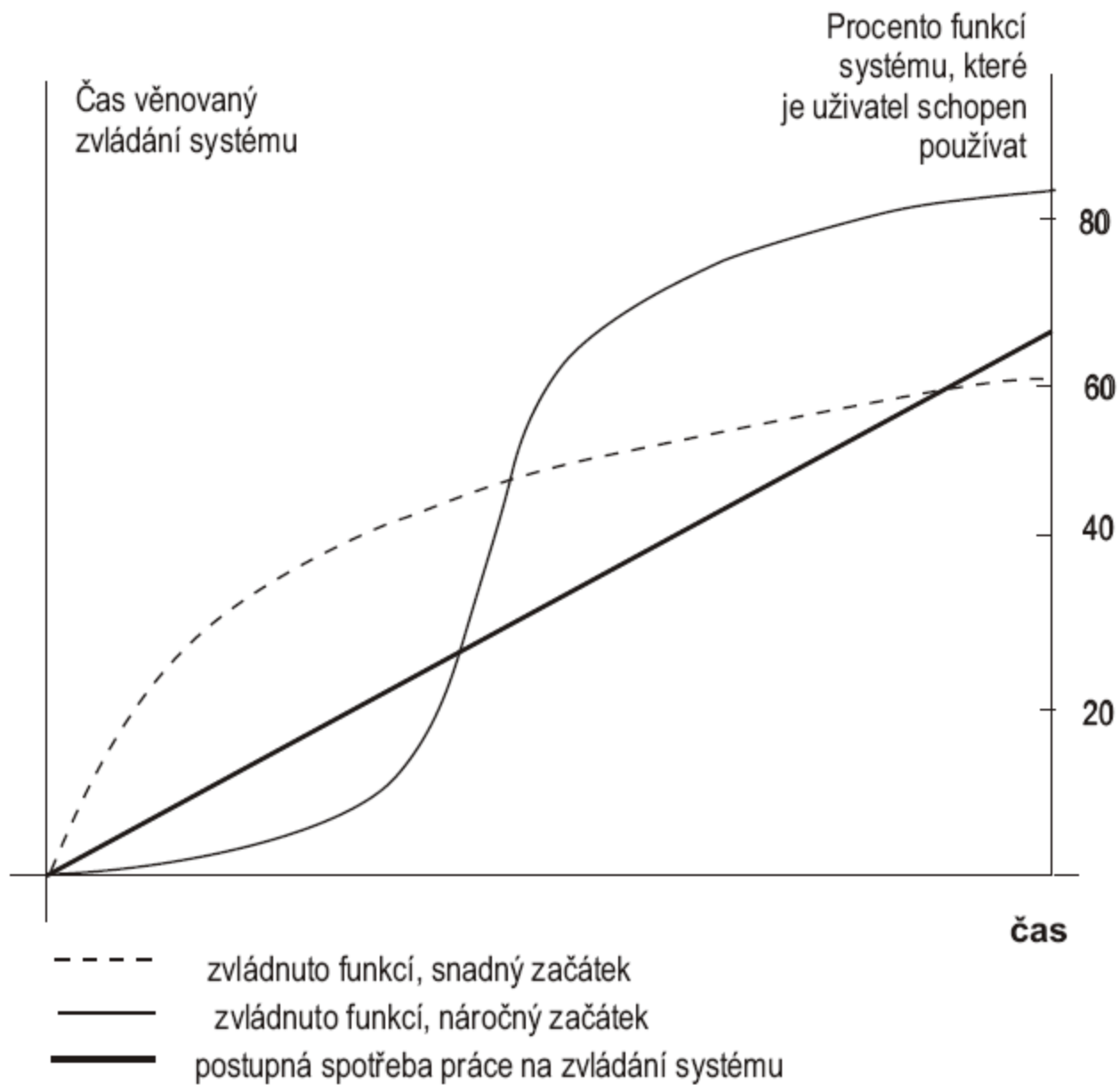
**Relativní počet
zvládnutých činností pro jednu osobu**

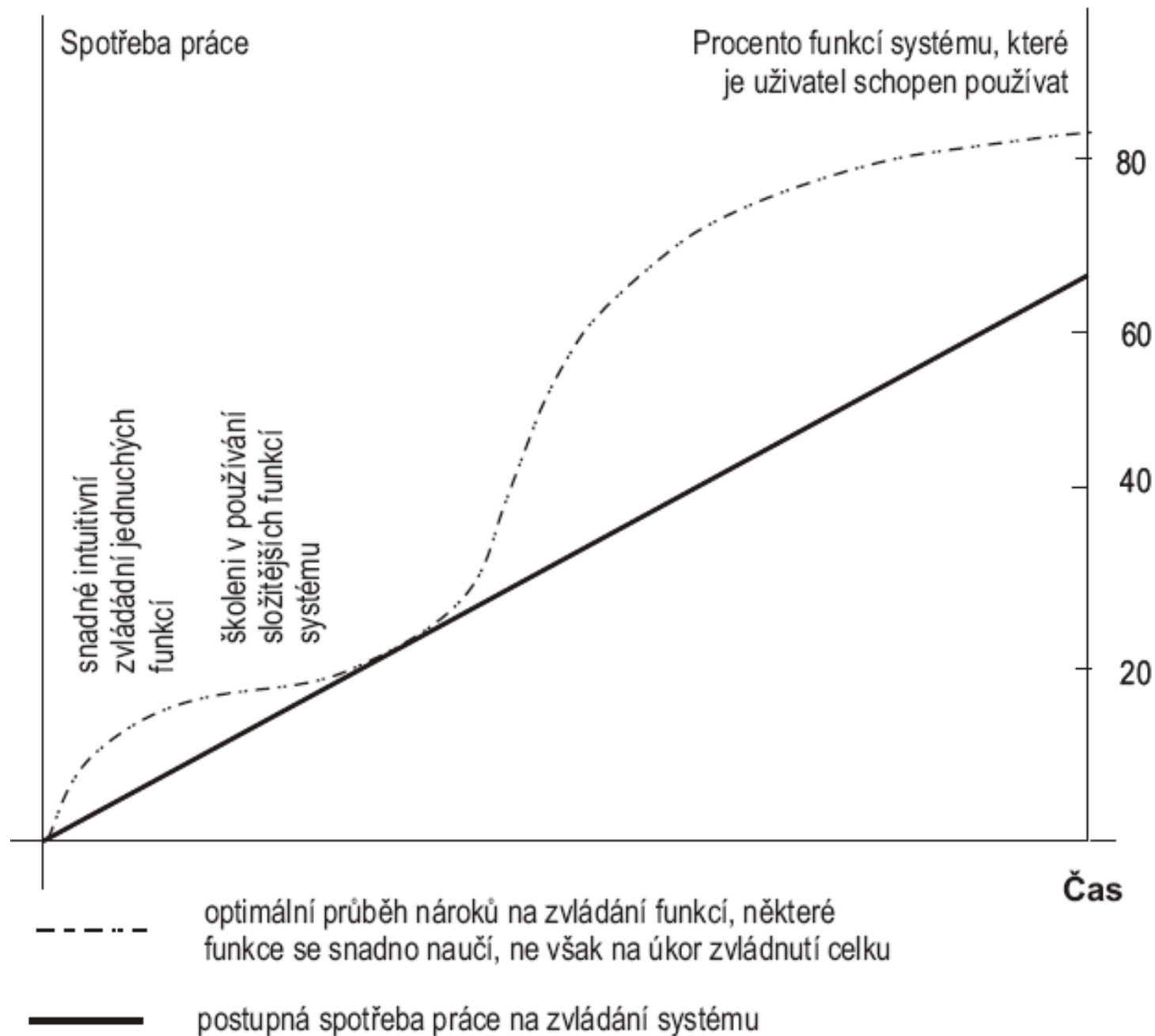


Učení metodou pokusů a omylů

Křivka učení a typy zvládnání







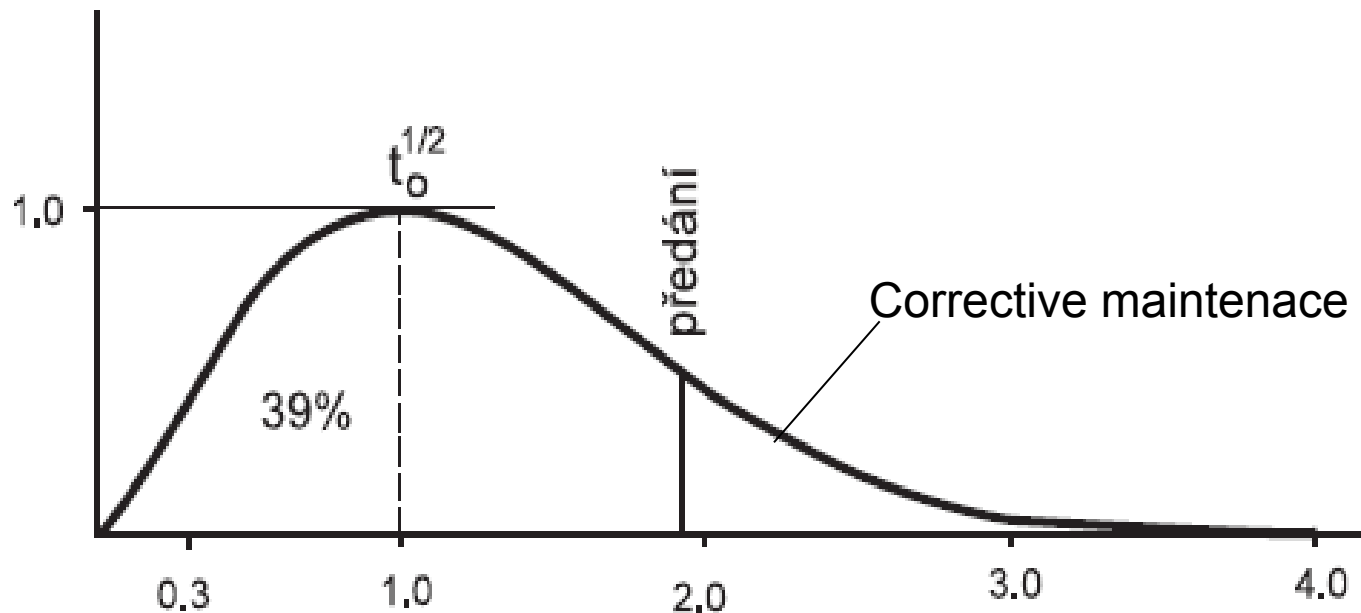
Údržba

I instrukce se z logického pohledu
(potřeb) opotřebují

Druhy údržby, opakování

- Corrective odstranění defektů, které nebyly detekovány oponenturami a testováním, vlastně ty, na které nebyl čas ani prostředky
- Enhance – vylepšování
- Adaptive – přizpůsobování změnám platformy, přenos, změny norem

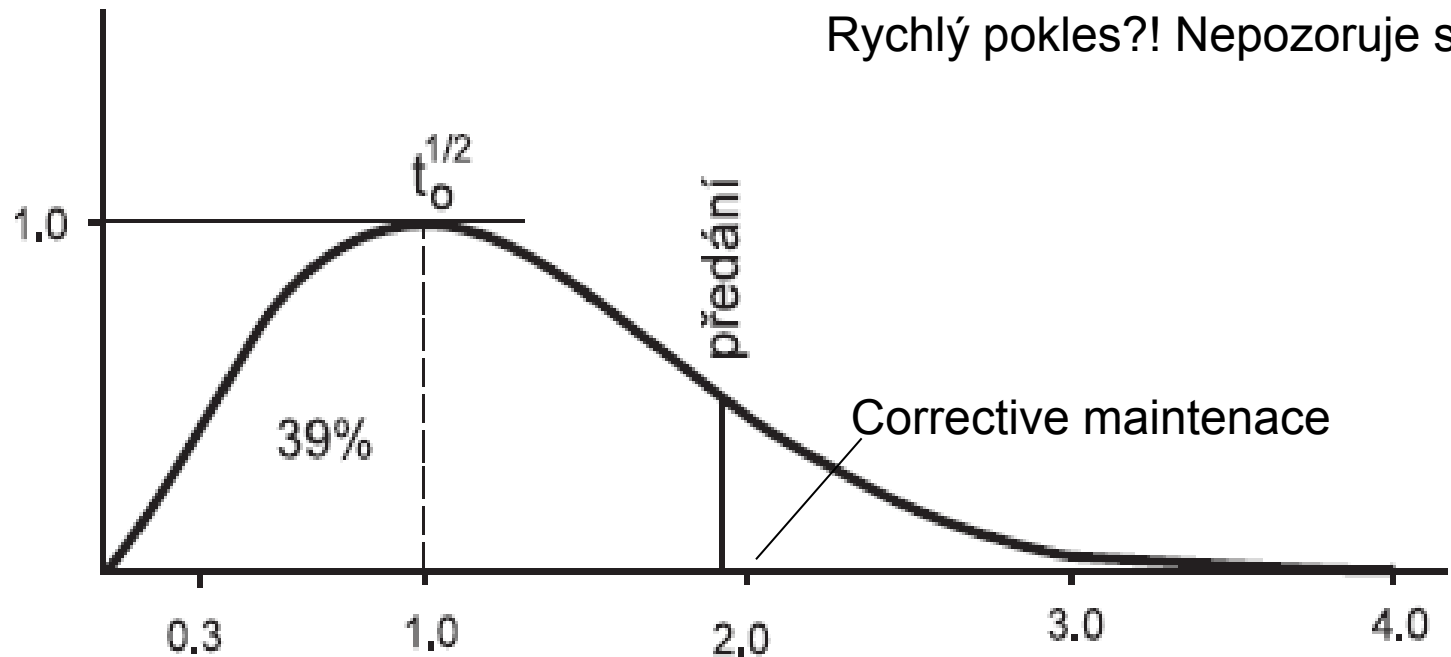
Model průběhu velikosti týmu a tedy intensity práce, starší varianta



Obr. 15.12: Rayleightova křivka. Část plochy pod křivkou po předání jsou práce, které budou provedeny v rámci údržby (corrective maintenance). To, co se do okamžiku předání nestihne udělat, přechází do údržby, kde se projevuje jako neodstraněné chyby.

$$team(t) \hat{=} K \cdot t/t_0 \cdot \exp\left(-\frac{t^2}{2t_0}\right)$$

Rychlý pokles?! Nepozoruje se



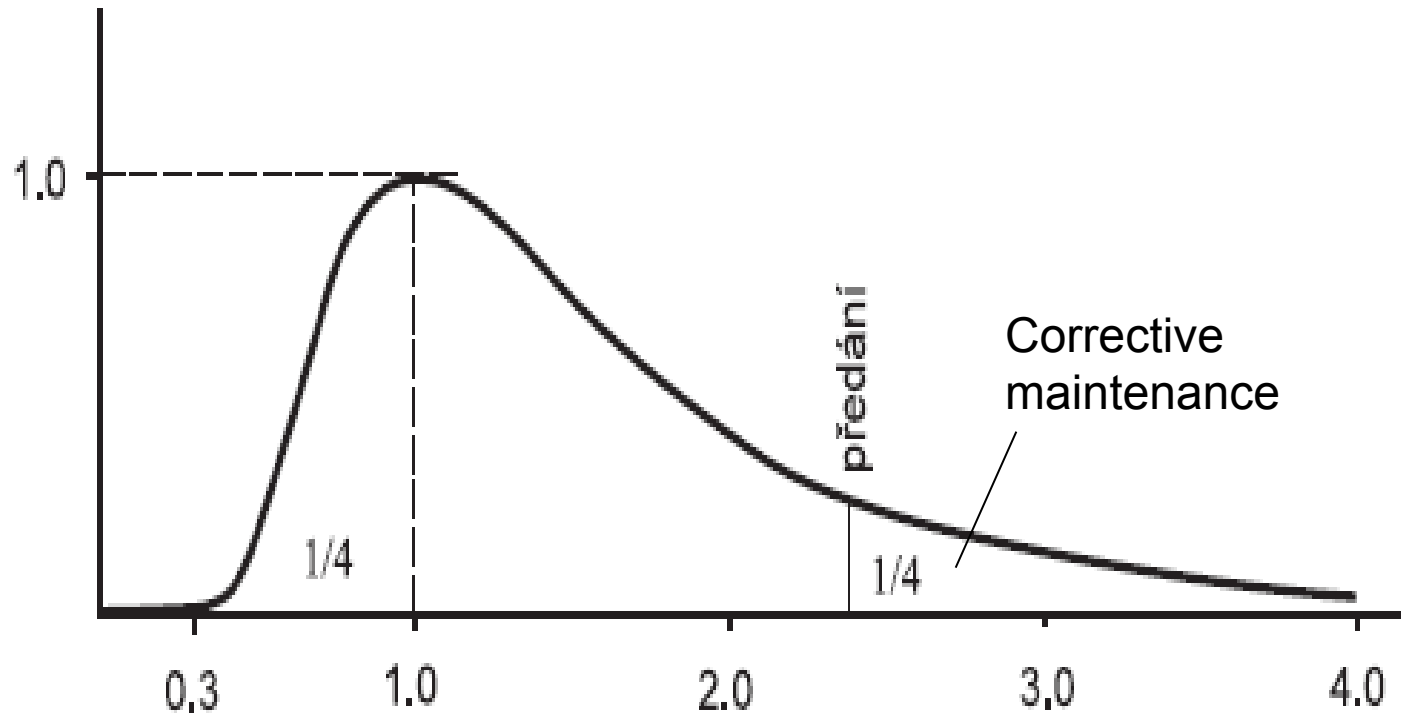
Obr. 15.12: Rayleightova křivka. Část plochy pod křivkou po předání jsou práce, které budou provedeny v rámci údržby (corrective maintenance). To, co se do okamžiku předání nestihne udělat, přechází do údržby, kde se projevuje jako neodstraněné chyby.

Pro $t > 3$ je velikost týmu prakticky nula \Rightarrow není co dělat, nebyla by corrective maintenance, to se nepozoruje. Pravidlo polovin: ve vrcholu jsem v půli se spotřebou práce i s dobou řešení, to je příliš optimistické

$$team(t) \hat{=} K \cdot t / t_0 \cdot \exp\left(-\frac{t^2}{2t_0}\right)$$

Jsou důvody pro komplikovanější model

Zobecnit na $t = aT+d$

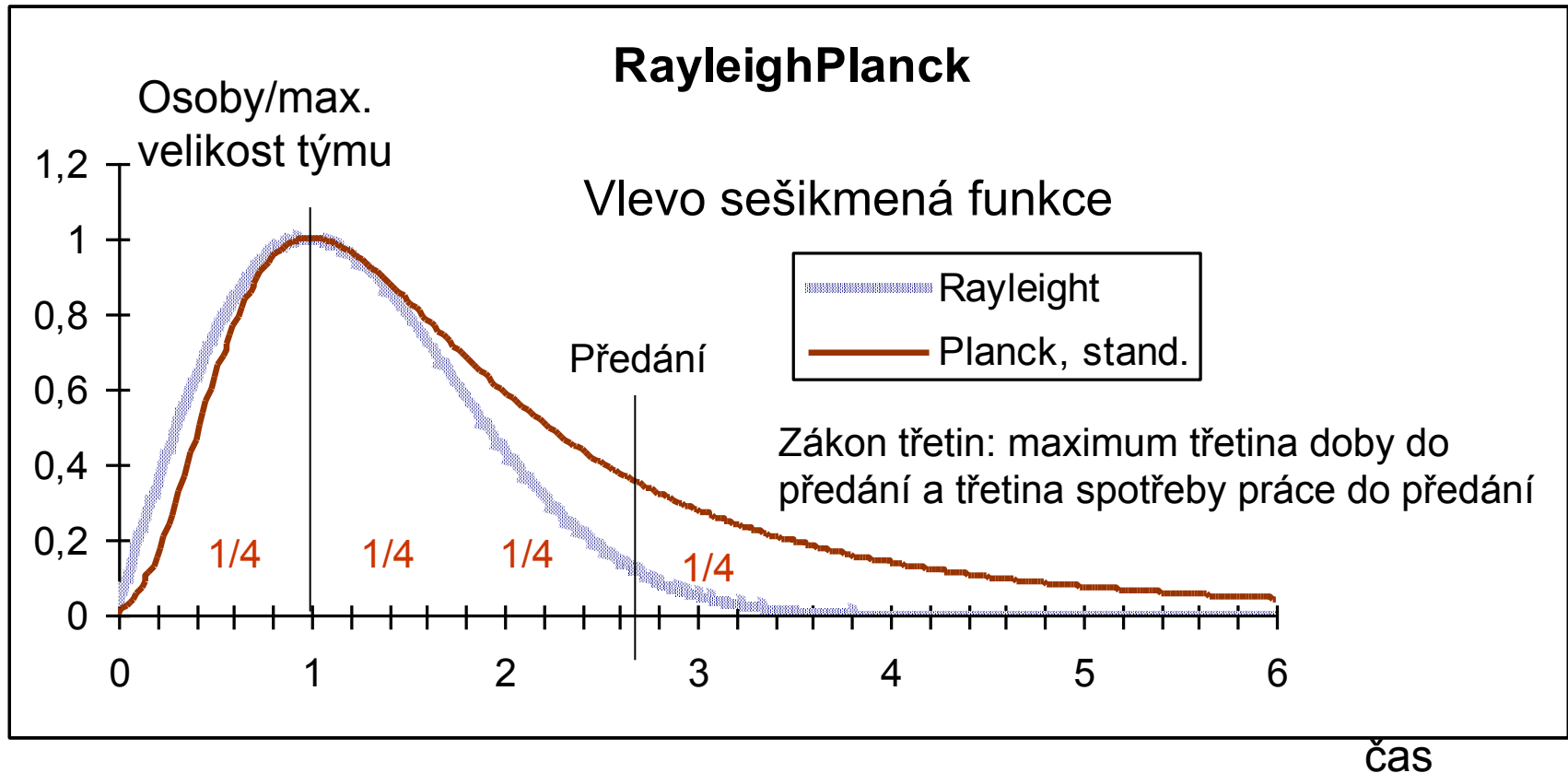


Obr. 15.13: Normalizovaný tvar Planckovy křivky $Planck(t) = 142.32 \cdot t^{-5} / (\exp(4.9651/t) - 1)$.

Mnoho práce se vykoná i pro $t > 5$,

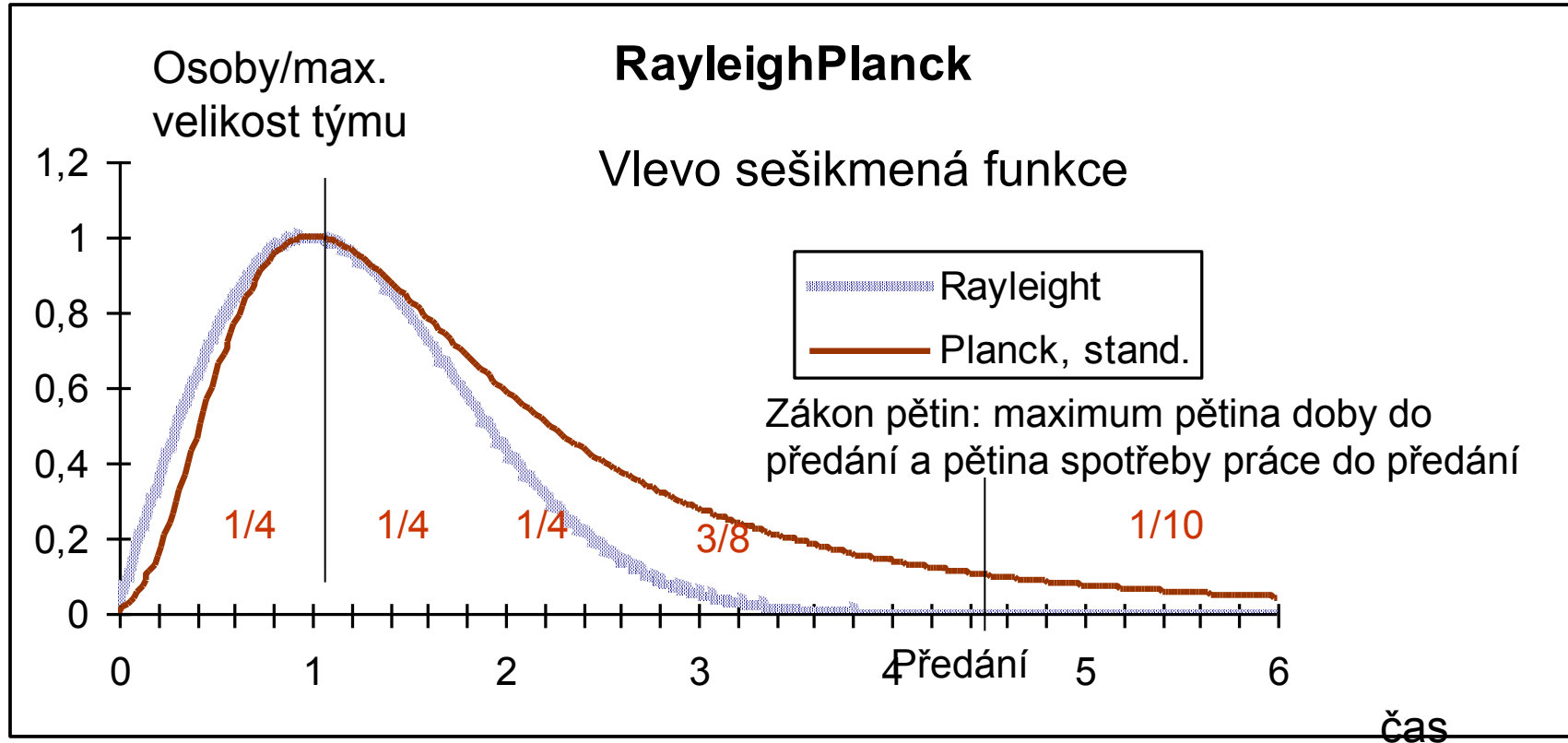
Posun vlevo lineární transformací nezávislé proměnné

Najímaný tým, vrchol a odhad doby řešení



Pravidlo třetin. Do vrcholu $1/3$ prací a $1/3$ doby (za běžných okolností). Transformace proměnných tak, aby max bylo v bodě 1 a mělo hodnotu 1 a v nule byla hodnota funkce prakticky nula. U pevného týmu odpovídá intenzitě práce

Najímaný tým, vrchol a odhad doby řešení krirických systémů

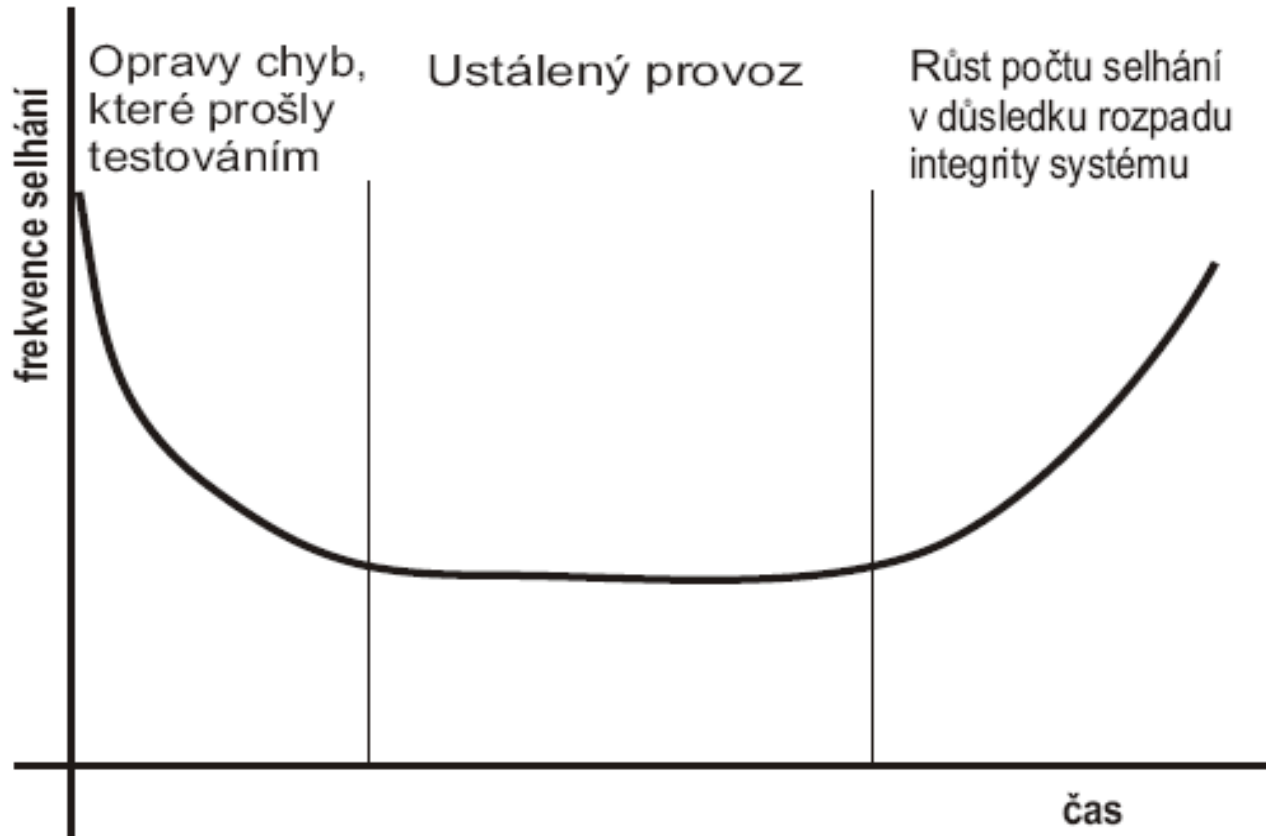


Ve vrcholu méně než $1/5$ doby do předání a málo více než $1/4$ (0.277) prací

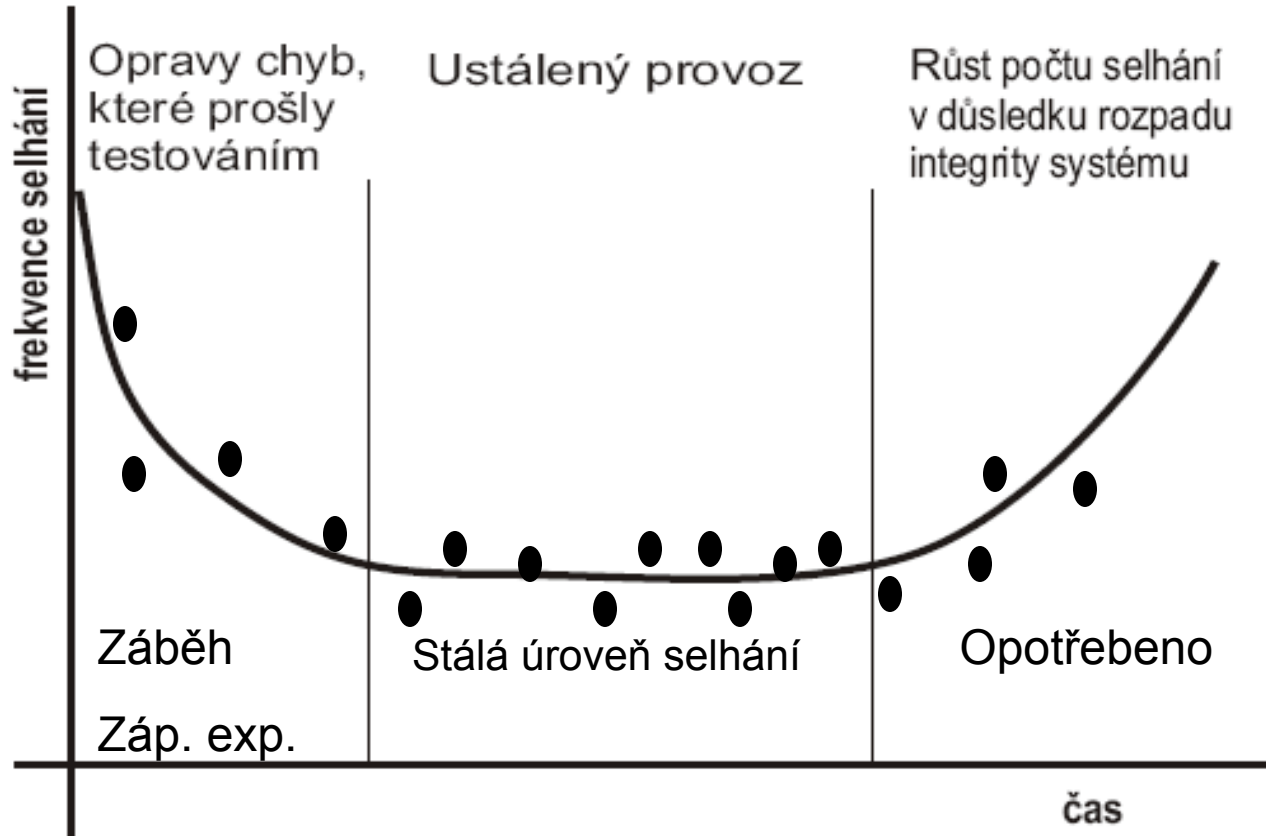
Etapy údržby

- Převzetí
- Etapa investic
 - Corrective maintenance, prvá vylepšení a přizpůsobení
- Etapa maximálního užitku
 - Vylepšení požadované uživateli
 - Regresní testy, stabilní provoz
- Zmenšování užitku
 - Vylepšení pro další uživatele, zlepšování výkonu, roste počet problémů

Vanová křivka. I SW se opotřebí



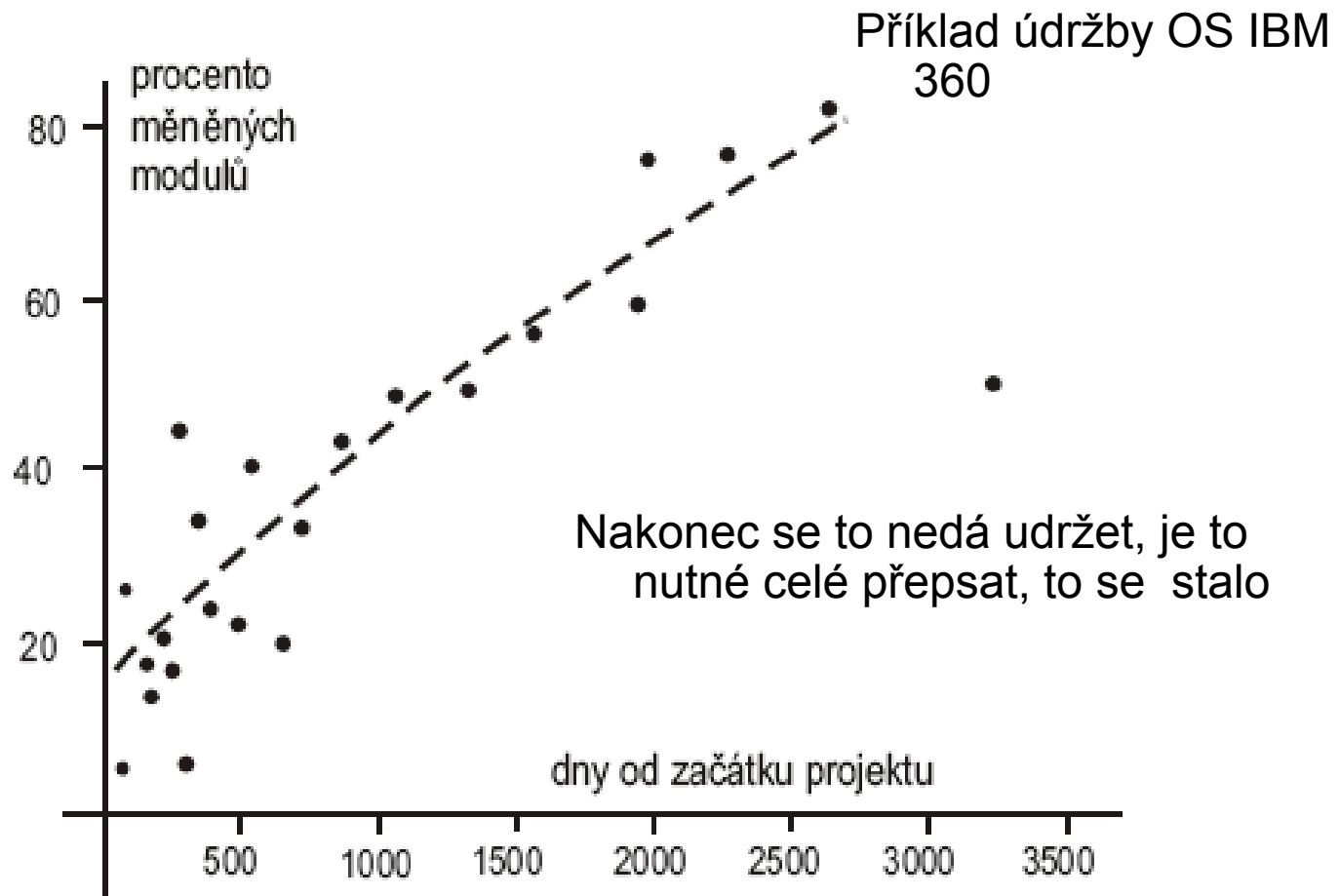
Vanová křivka. I SW se opotřebí



Dno vanové křivky implikuje **nenulovou** střední frekvenci selhání

- Opravou se zanesou další chyby
- SW se stále upravuje (vylepšuje, přenáší na nové platformy) a tím se zanáší problémy a další chyby, tím se údržba stává stále pracnějšší.
 - Za odstraněný defekt přibude často nový, někdy i více než jeden
 - Není proto možné zero defect software

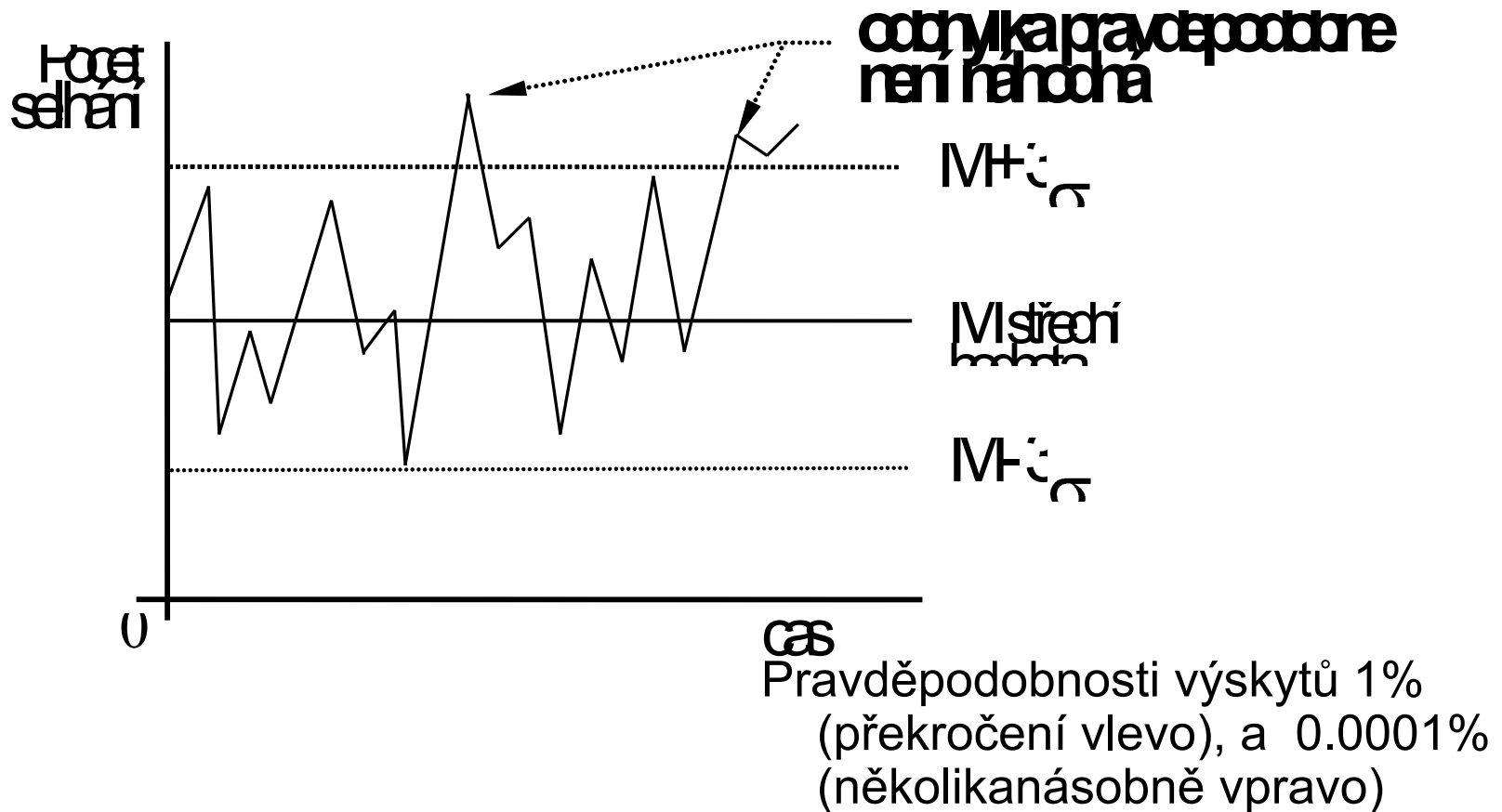
Složitost údržby roste s časem, důvod stoupající větve vanové křivky



Hlubší důvod k vyhození systému

- Pozoruje se, že si systém zachovává své základní vlastnosti plynoucí z filosofie volby požadavků a technologie návrhu a vývoje
- Nectnosti systému se spíše zviditelňují a zesilují filosofie zastarává, není „in“,
- Platí to i pro jiné technické prostředky (nákladní auta)

Jednoduchá kontrola toho, zda došlo k opotřebení



Co se tím vlastně prokazuje

- Je málo pravděpodobné ($p \ll 0,001$), že *nedošlo* k posunu střední hodnoty směrem nahoru, tj. je pravděpodobné, že k němu došlo (tak se testují statistické hypotézy)
- Lze použít efektivnější prostředky mat. statistiky, např. *t* test
 - je účinnější a přesnější, není ale tak názorný a je proto méně vhodný pro on-line zásahy

Co snižuje pracnost údržby

- Správné specifikace (správné požadavky ale také správná dekompozice)
- Vhodná architektura celku (SOA), Struktura částí
(Objektová orientace, komponenty)
- Převzetí existujícího (knihovny, produkty třetích stran existující domácí SW)
- Správné procesy vývoje (inkrementálnost, agilnost) a kvalita dokumentace, kvalita oponentur a testů, normy
- Přenositelné technologie (Java), normy, PaaS
- Vývojové nástroje (menší rozsah dokumentů, srozumitelnost, podpora korektnosti oprav,...)

Co snižuje pracnost údržby

- Dobře vedené programování (Gunderloy)
 - Používám, co je napsáno (existující aplikace, produkty třetích stran, knihovny), normy
 - Používám moderní metodiky (OO, SOA, metanástroje jako XML a s ním spojené jazyky a DB) a vhodný jazyk
 - Agilní metody vývoje
 - Dohodnuté standardy
 - Komentáře, volby identifikátorů, pravidla strukturování, oponování, vývoj testů,

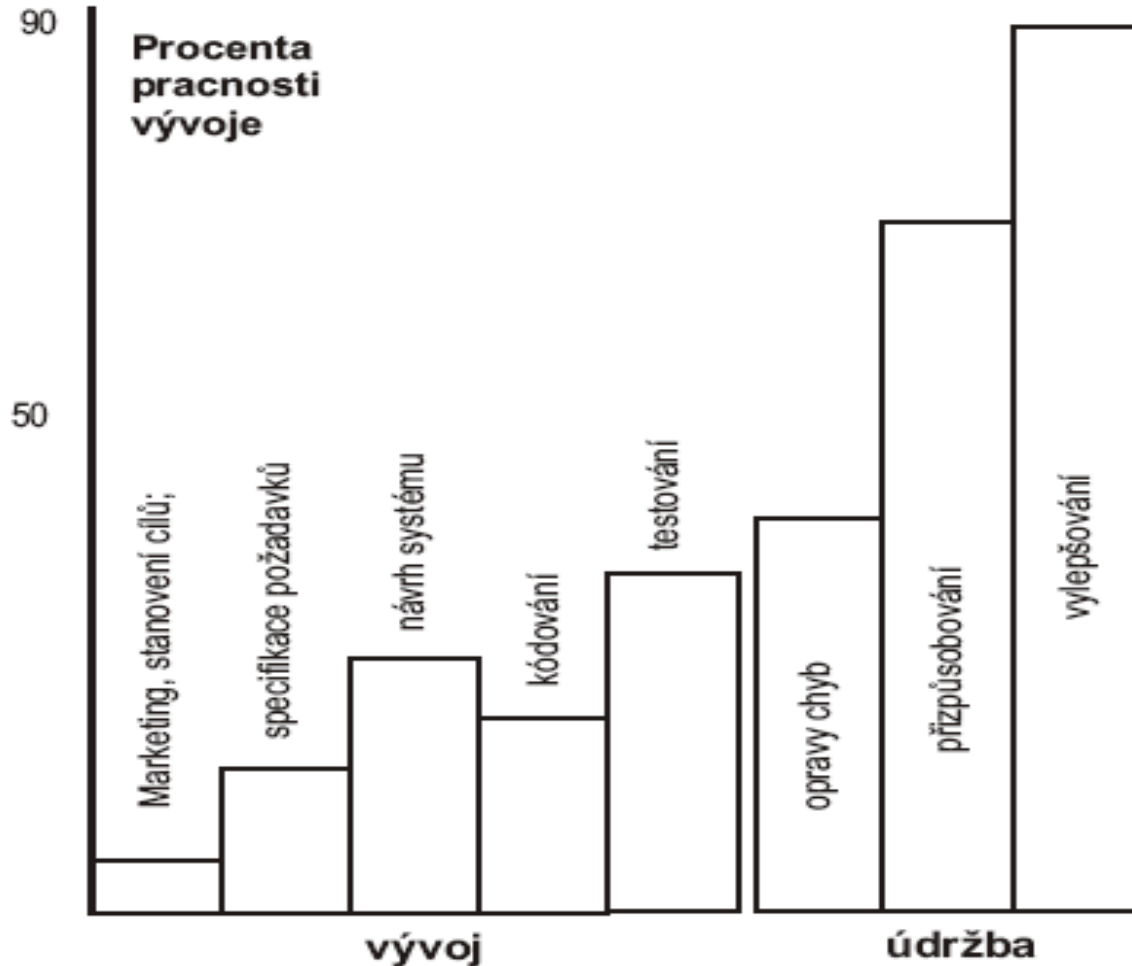
Co snižuje pracnost údržby

- Kvalita technik údržby
 - Automatizace opakování testů
 - DB reklamací a defektů
 - Sledování trendů defektů (frekvence)
 - Sledování prapříčin, jaká chyba člověka je prvotní a které chyby jsou důsledkem
 - Použití logů komunikace přes uživatelské rozhraní a mezi komponentami (výhodné v SOA) a rozhraní

Reinženýring

- V podstatě kompletní přepsání systému jako jediná alternativa k jeho zrušení
- Varianty
 - Enhansive, obvykle s novými funkcemi, nová architektura
 - Adaptive, změna platformy a jazyka či databáze
 - Vyjíměčně větší spolehlivost (SLA)
- Rizika: ovlivnění starými praktikami, neochota k žádoucím změnám, riziko, že to nestojí za to

Podíly pracnosti



Údržba stojí podstatně více než vývoj (obvykle dvakrát více, to závisí na počtu uživatelů).

U customizovaných systémů je podíl údržby pro jednotlivé instalace podstatně menší (platí se tím, že se zákazník musí přizpůsobovat

Jaké techniky snižují pracnost údržby

- Vždy: Kvalitní architektura, kvalitní specifikace, kvalitní vývojové prostředí
- Corrective
 - Architektura SW, možnost agility
 - Moderní metody vývoje moderní vývojová prostředí, vodný jazyk
- Adaptive
 - PaaS, nástroje nezávislé na platformě
- Enhansive
 - SOA s hrubozrnným rozhraním, automatické testy, kvalitní specifikace

Jak se projeví SaaS, software as a service

A také cloud computing

Platform as a service

Zatím ve fázi líbánek