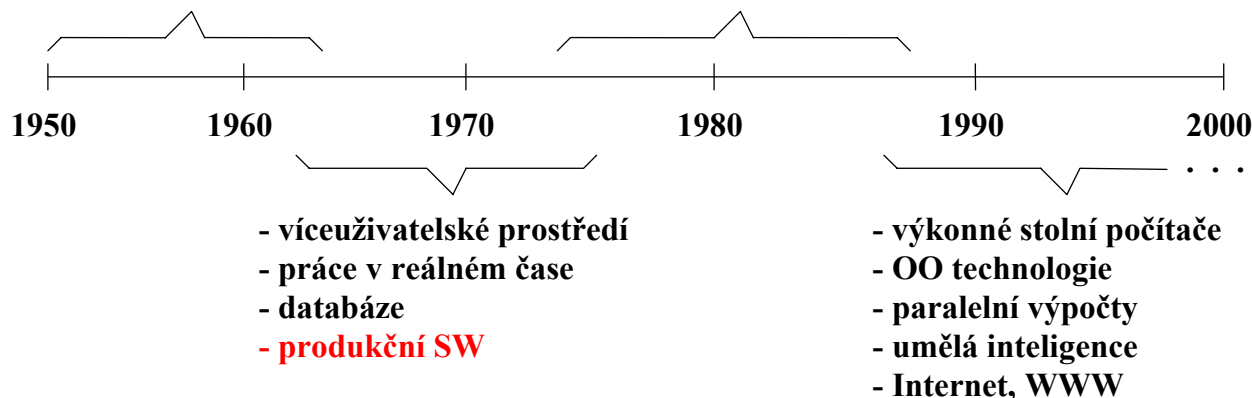


1 Úvod

1.1 Vlastnosti programového vybavení (SW)

- dávkové zpracování
- omezená distribuce
- zakázkový SW
- distribuované systémy
- vestavěná „intelligence“
- laciný HW
- vliv zákazníka



I - málo systematických metod, kdo SW vyvinul, ten i udržoval

II - růst výkonnosti HW, **SW jako výrobek**, problém údržby, **krize SW**

III - PC a WS, růst počtu distribucí, prodej SW předbíhá prodej HW

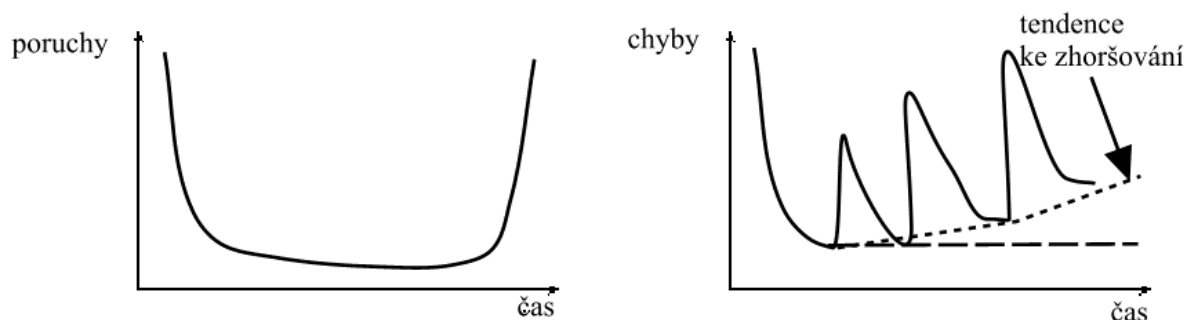
IV - zesilují problémy spojené se SW (schopnost využít potenciálu HW, růst požadavků na nové programy, neschopnost údržby)

• Programové vybavení zahrnuje:

- počítačové programy,
- datové struktury,
- dokumentace.

• Vlastnosti SW:

- vyvíjí se, není vyráběn v klasickém smyslu,
- neopotřebovává se (ale zhoršuje), neexistují „náhradní díly“



- je spíše budovaný na zakázku než sestavovaný z existujících dílů
- hledání metod *opakovaného použití* (reusability) - knihovny, OO přístup, technologie založené na komponentách

- **Generace programovacích jazyků**
 - strojově-orientované (2G),
 - vyšší programovací jazyky (3G),
 - neprocedurální jazyky (4G).
- **Typy programů**
 - různá hlediska klasifikace, možné členění:
 - Systémový SW** - slouží jiným programům (jádro operačního systému, překladač, ...).
 - SW pro práci v reálném čase** - typicky sběr dat+analýza+řízení, rozhodující je časové omezení odezvy (řízení technologických procesů).
 - Firemní programy (business software)** - interakční, víceuživatelské, databázová technologie, velké objemy dat. Přejít od jednotlivých programů k integrovaným systémům, součástí nebo nadstavbou často programy a systémy na podporu rozhodování.
 - SW pro VTV** - programy na podporu složitých výpočtů různých vědních oborů (předpověď počasí, návrh plošných spojů, ...).
 - SW z oblasti umělé inteligence** - nenumerické algoritmy k řešení složitých problémů (expertní systémy, rozpoznávání, neuronové sítě, ...).
 - dávkové/interakční - s grafickým/textovým rozhraním, ...

- **Krise SW**
 - 60. léta - růst výkonnosti HW → požadavky na rozsáhlé systémy
 - techniky návrhu malých systémů, špatné odhady, špatně udržovatelné programy, slabá výkonnost
 - ⇒ SW v krizi, cena HW ↓, cena SW ↑
 - ⇒ potřeba lepších metod, technik a nástrojů (obsah pojmu softwarové inženýrství).
 - **Problémy:**
 1. špatné odhady doby řešení a nákladů (plánování a řízení projektu)
 2. nedostatečná produktivita vývojářů SW (nástroje)
 3. nízká kvalita programů (význam systematických opatření k zajištění kvality)
 4. obtížná udržovatelnost (dokumentace, CASE).
 - **Příčiny:**
 - v povaze SW - kvalita dána kvalitou jediného kusu i při velkých distribucích, odstraňování chyb
 - v lidech - nekompetentní (v oblasti SW) manažeři, konzervativní praktici, rozptýl produktivity programátorů (1:28), problémy v komunikaci vývojář - zákazník

- **SW mýty:**

mýty managementu: tlak rozpočtu, času, kvality

„Mojí lidé přece mají moderní vývojové nástroje; kupujeme jim nejnovější počítače.“

- rozhodující význam výkonného vývojového prostředí (SW)

„Jestli se dostaneme s projektem do skluzu, přijmeme další programátory a doženeme to.“

- Brooksův zákon:

„Přidání lidí ke zpožděnému SW projektu zvýší jeho zpoždění.“

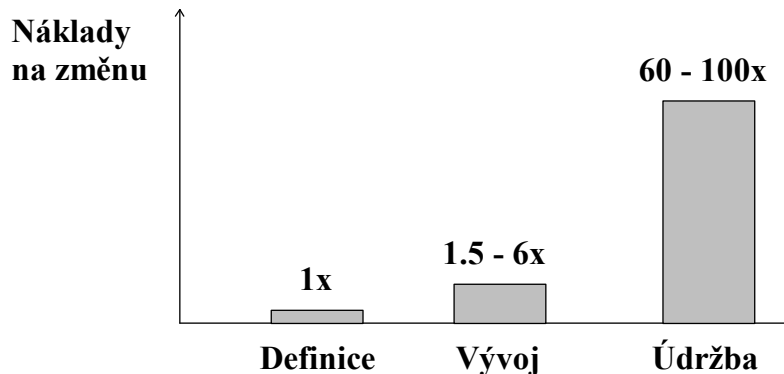
mýty zákazníka: chybná očekávání → nespokojenost s vývojářem

„Obecná specifikace cílů je dostatečná k tomu, aby se začaly psát programy.“

- formální a detailní specifikace je podstatná → význam komunikace zákazník-vývojář

„Požadavky na projekt se neustále mění, ale změnám se lze snadno přizpůsobit, protože SW je pružný.“

- vliv změn je závislý na fázi zavedení



mýty praktika:

„Náš úkol skončí, jakmile napíšeme program a uvedeme ho do provozu.“

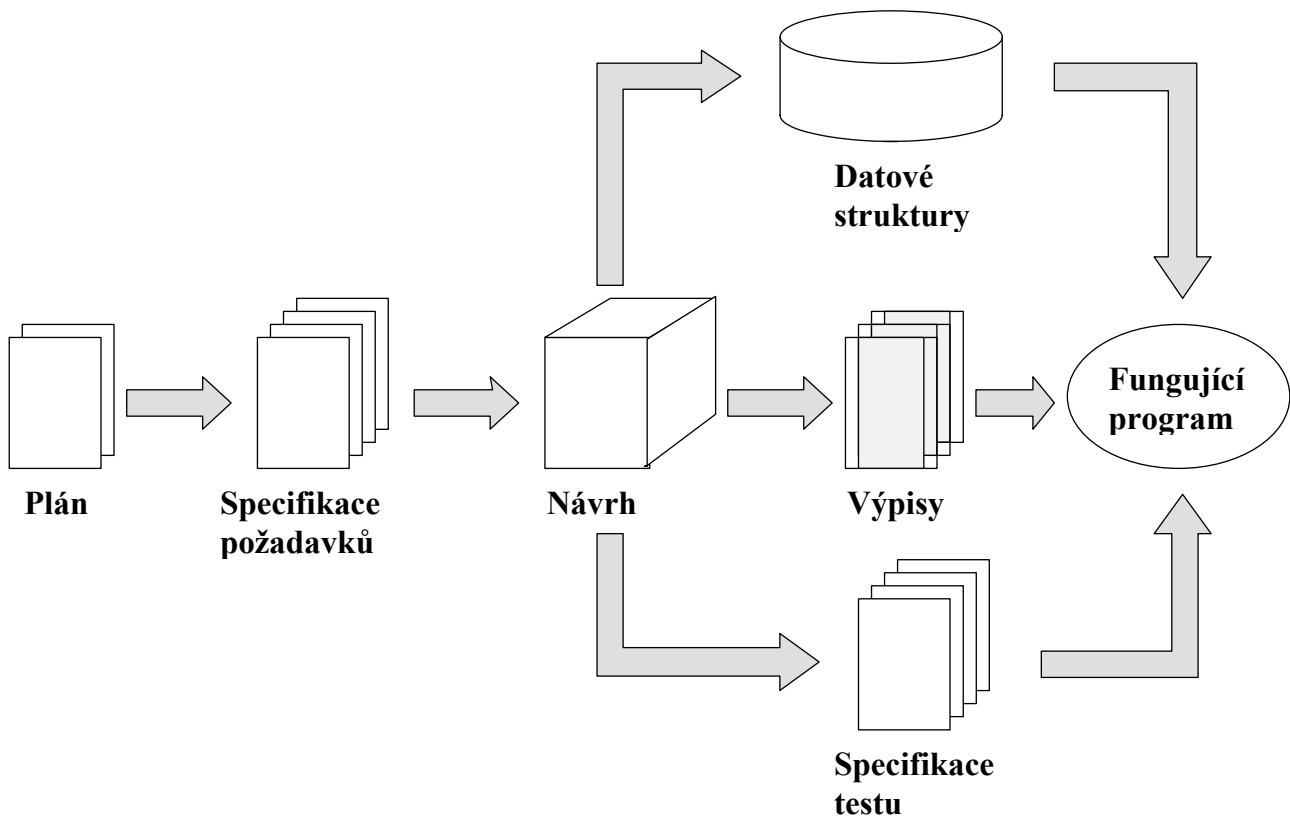
- 50 - 70% úsilí bude spotřebováno po prvním dodání programu zákazníkovi

„Dokud neuvedu program do chodu, nemám možnost posoudit jeho kvalitu.“

- techniky posuzování a oponentur lze aplikovat už v počátcích projektu

„Jediným výsledkem úspěšného projektu je fungující program.“

- fungující program je pouze část tzv. SW konfigurace



1.2 Předmět softwarového inženýrství

Podmínky formování disciplíny - krize SW

Fritz Bauer (1969 konference sponzorovaná vědeckým výborem NATO k problematice softwarového inženýrství):

„Stanovení a používání spolehlivých inženýrských principů za účelem získat ekonomicky SW, který je spolehlivý a pracuje efektivně na reálných počítačích.“

Systematický přístup k vývoji SW, provozování, údržbě a nahrazování starého SW.

- **Zahrnuje:**

- metody
 - nástroje
 - procedury
-
- pro manažera podpora řízení vývoje
 - pro praktika základ efektivního vývoje kvalitních a udržitelných programů

Metoda - „Jak na to“ - plánování a odhady projektu, analýza požadavků, návrh datových struktur, struktury programu, kódování, testování, údržba → notace + kritéria vyhodnocení kvality.

Nástroj - automatizovaná/poloautomatizovaná podpora metod
CASE - integrované nástroje (HW, SW, databáze).

Procedura (metodologie, metodika) - posloupnost aplikace metod a požadovaných výstupů (deliverables), řídicích kroků k zajištění kvality a koordinace, milníky

- Softwarové inženýrství se zabývá programovými systémy vyvíjenými týmem
⇒ technické + netechnické aspekty

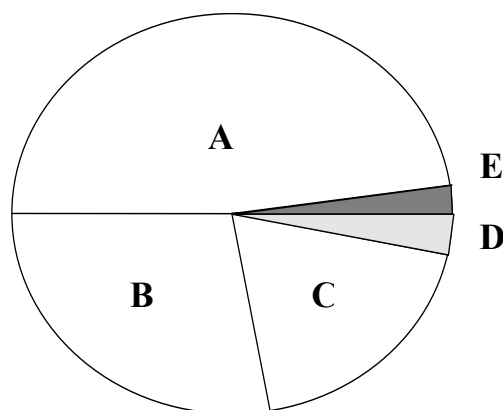
Softwarové inženýrství učí, jak se vyvarovat budoucích chyb, ne jak je řešit.

- softwarové inženýrství jako disciplína a jako proces

Čím se vyznačuje dobrý SW?

? „Pracuje, jak chce uživatel.“ ?

funkčnost + udržovatelnost + spolehlivost + efektivní + vhodné uživatelské rozhraní



A - dodán, nikdy úspěšně nepoužit (\$3.2 mil. - 47%)

B - zaplaceno, ale nedodáno (\$1.95 mil. - 29%)

C - použit s rozsáhlými změnami nebo později nepoužit (\$1.3 mil. - 19%)

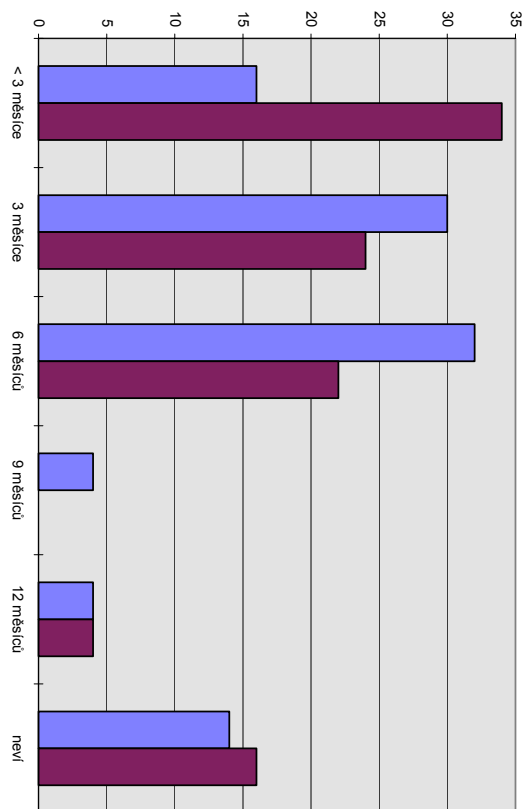
D - použit po úpravách (\$0.198 mil. - 3%)

E - použit beze změny (\$0.119 mil. - 2%)

- Delloite & Touche - celosvětový průzkum trhu s informačními technologiemi (1996):
- oblast aplikací

oblast	zakázky [%]	komerční [%]
svět	43	57
ČR	44	56

- Zkracování doby mezi uváděním nových verzí



Činnosti při údržbě programů:

