

# PROBLÉMY A SPECIFIKA VÝVOJE SOFTWARE

Vývoj prvních programů byl prováděn nadšenci, programy byly šité na míru. Žádná metodika vývoje SW v té době neexistuje.

Vývoj SW byl vnímán jako výzkum. Cíl, co bude software dělat, si určoval programátor sám na základě velmi vágního zadání (ostatní často neměli představu, co by od počítačů a software mohli chtít). Vývoj většinou nebyl omezen žádnými termíny (až byl software hotov, došlo k jeho nasazení).

V důsledku stále většího rozmachu informatiky a počtu projektů, které neskončily úspěšně (ve smyslu – nebyly dokončeny včas nebo byly příliš nákladné nebo nebyly dokončeny vůbec). Na přelomu 60. a 70. let 20. století se začalo mluvit o tzv. „softwarové krizi“.

## SOFTWAREOVÁ KRIZE

- Neúnosné prodražování projektů
- Neúnosné prodlužování projektů
- Nízká kvalita programů
- Nízká produktivita programátorů
- Neefektivita vývoje
- Nejistota výsledku

„Softwarová krize“ vyvolala potřebu používání metodik pro řízení vývoje software (IS).

Jak vypadají problémy softwarové krize z dnešního pohledu?

### *PROBLÉM – ŠPATNÁ KOMUNIKACE*

Jedním z hlavních problémů je špatná komunikace. Dříve často zákazník jednal přímo s programátorem.

Dnes: tendence k oddělení funkce analytika od vývojáře (kodéra).

### *PROBLÉM – NESPRÁVNÝ PŘÍSTUP*

Problém přístupu lidí k vývoji. Programátoři občas sklouzávají k tendenci předvést se, seberealizovat, „vyřádit se“.

Dnes: zaměření na týmovou spolupráci, důležitá je spokojenost zákazníka.

### *PROBLÉM – ŠPATNÉ PLÁNOVÁNÍ*

Je obtížné vypracovat plán vývoje, který je přijatelný pro zákazníka a realizovatelný pro vývojáře.

Představa typu „nějak se to stihne“.

Po zadání projektu někdy následovalo bezprostřední „bušení do klávesnice“.

Dnes: metodiky vývoje software

### *PROBLÉM – NÍZKÁ PRODUKTIVITA*

Programátoři se zabývali vším možným jen ne tím, co bylo potřeba. Tendence psát kód okamžitě, vychrlit čím jak nejvíce řádků kódu.

Dnes: tým s přidělenými rolemi. Důraz na koordinaci vývoje. Vývoj podle předem stanovených specifikací.

### *PROBLÉM: PODCENĚNÍ HROZEB A RIZIK*

Problémy, které by mohly být vyřešeny hned na začátku, byly přehlíženy. „To vyřešíme nakonec“, „to nebude problém“, „to se nepozná“.

Dnes: snaha odhalit chyby na začátku. Metodiky na provádění analýzy rizik (rozbory potenciálních hrozeb). Metodiky, které berou riziko jako základní jednotku.

## ODLIŠNOST VÝVOJE SW OPROTI OSTATNÍM OBORŮM

Čím je vývoj softwaru specifický oproti ostatním oborům?

*KVALITA ODBORNÝCH PRACOVNÍKŮ JE PRO ÚSPĚCH ROZHODUJÍCÍ.*

V klasických oborech je strůjcem úspěchu management, pracovníci na nižší úrovni jsou považováni za snadno zastupitelné a nahraditelné. V oblasti vývoje software to tak úplně neplatí a odchod klíčového programátora z týmu může způsobit vážné komplikace (nebo dokonce krach projektu).

Velký rozdíl je v produktivitě. Programování **není jen o zvládnutí programovacího jazyka, ale i o zkušenostech**. Mezi programátory jsou zásadní rozdíly dané vzděláním, zkušenostmi, talentem.

*ODBORNÍ PRACOVNÍCI SE MUSÍ NEUSTÁLE ZDOKONALOVAT A VZDĚLÁVAT.*

Nové technologie přicházejí **cca co tři roky**.

Neustálé školení a vzdělávání je nutnost. Programátorům hrozí technologické zaostání a na druhé straně, díky rychlému tempu, také vyhoření.

*VÝHODNÝ JE JINÝ ZPŮSOB PRÁCE.*

Programování není o osmihodinové pracovní době.

Programátory často práce baví a je výhodné pracovní dobu pružně měnit. V týmu prodchnutém nadšením lze dosáhnout značné produktivity, nesmí ale jít o práci ve stresu a období zvýšeného úsilí musí být časově omezeno.

*NEJPRODUKTIVNĚJŠÍ JSOU MALÉ, POZITIVNĚ MOTIVOVANÉ TÝMY*

Striktně hierarchický systém, s prací pod tlakem, že nebudou prémie, v softwarovém vývoji nefunguje.

Vývoj software je tvůrčí práce. Manažer vývoje by neměl vývoj řídit svými příkazy, jestliže tým úspěšně funguje. Při vzniku problémů by naopak měl zasáhnout čím jak nejdříve.

*CHYBA JE ZÁKONITOU SOUČÁSTÍ VÝVOJOVÉHO PROCESU*

Vývoj SW je tvůrčí a kreativní činnost, nelze ji naplánovat jako stavbu budovy.

I v nejlépe řízených projektech se vyskytnou chyby. Je ale třeba stanovit takové postupy, aby se chyby maximálně eliminovaly.

## PRUDKÝ TECHNOLOGICKÝ ROZVOJ PŘINÁŠÍ NOVÉ PŘÍLEŽITOSTI

Rutiněři na vedoucích místech je často neumí rozpoznat. Iniciativa může přicházet zdola. Úspěchy v minulosti nic neznamenají

Říká vám něco jméno Digital Equipment Corporation (DEC)?

(V 20. století spolu s IBM největší IT firma na světě, přes 50 tisíc zaměstnanců, jeden z leaderů na poli mainframů, firma, která přinesla úspěšné řady mainframe počítačů, jako byly PDP, VAX, ALPHA, operační systém RSX a VMS. Dnes je to jen malá divize v koncernu Hewlett Packard.)

## NEJČASTĚJŠÍ PROBLÉMY VÝVOJE SW

- Zpoždění
- Vysoká chybovost
- Neplnění požadované funkčnosti
- Nedostatečná výkonnost
- Složité uživatelské rozhraní
- Obtížná udržitelnost programu

## ZÁKLADNÍ PŘÍČINY PROBLÉMŮ

- Podcenění projektu a špatný odhad (čas, náklady)
- Špatné zadání
- Nedostatečná analýza
- Přílišná složitost projektu
- Přehnaný důraz na technologii (použití novinek bez zkušeností)
- Špatná kvalita programového kódu (chybový, nesrozumitelný, pomalý, nedostatečně komentovaný)
- Nevhodné metodiky, postupy, technologie
- Nedostatečné testování
- Špatné projektové řízení