



národní
úložiště
šedé
literatury

Návod na stanovení požadavků na programové vybavení

Šebesta, Václav
1996

Dostupný z <http://www.nusl.cz/ntk/nusl-33676>

Dílo je chráněno podle autorského zákona č. 121/2000 Sb.

Tento dokument byl stažen z Národního úložiště šedé literatury (NUŠL).

Datum stažení: 24.04.2024

Další dokumenty můžete najít prostřednictvím vyhledávacího rozhraní nusl.cz .

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Návod na stanovení požadavků na programové
vybavení

Václav Šebesta

Technical report No. 692

October 1996

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic
phone: (+422) 6605 2060 fax: (+422) 8585789
e-mail: vasek@uivt.cas.cz

Návod na stanovení požadavků na programové vybavení ¹

Václav Šebesta

Technical report No. 692
October 1996

Abstrakt

Podle tohoto návodu je možno stanovovat požadavky na programové vybavení. Popisuje nezbytný obsah a jednotlivé položky správných požadavků na programové vybavení (dále jen PPV) a obsahuje i nástin prototypu PPV. Návod je napsán proto, že současný stav v této oblasti dostatečně nepodporuje standardní formální dokumenty. Návod je aplikovatelný pro vnitřní a komerční programové produkty. Cílem tohoto návodu je specifikace požadavků na programové vybavení, které má být vyvinuto. Aplikace tohoto materiálu na dříve vypracované programy je nesprávná.

Keywords

programové vybavení, řízení kvality, požadavky na programy

¹Podle normy ANSI/IEEE 830-1984

Obsah

| | |
|--|----|
| Definice | 1 |
| 1 Požadavky na programové vybavení (PPV). | 1 |
| 1.1 Jednoznačnost. | 2 |
| 1.2 Úplnost. | 3 |
| 1.3 Ověřitelnost. | 4 |
| 1.4 Konzistence. | 4 |
| 1.5 Modifikovatelnost. | 5 |
| 1.6 Sledovatelnost. | 5 |
| 1.7 Použitelnost ve fázi využívání a údržby. | 6 |
| 2 Společná příprava PPV. | 6 |
| 3 Nástroje pro vývoj PPV. | 7 |
| 4 Požadavky na programové vybavení. | 7 |
| 5 Rámcový prototyp PPV. | 8 |
| 6 Vysvětlivky k rámcovému prototypu | 10 |
| 6.1 Úvod (Kapitola 1.) | 10 |
| 6.2 Celkový popis (Kapitola 2.) | 11 |
| 6.3 Stanovené požadavky (Kapitola 3.) | 14 |

Definice.

Smlouva (contract). Zákonný závazný dokument, sjednaný mezi dodavatelem a odběratelem (zákazníkem). Obsahuje technické, organizační, cenové a další požadavky na výrobek.

Zákazník (customer). Osoba nebo osoby, které platí za výrobek a obvykle (ale ne nezbytně) rozhodují o požadavcích na výrobek. Ve smyslu tohoto návodu mohou být zákazník i dodavatel členy téže organizace.

Jazyk (language). Prostředek komunikace s vlastní syntaxí a sémantikou, skládající se z množiny reprezentací, konvencí a nezbytných pravidel, využívaný pro přenášení informací.

Rozdělování (partitioning). Dělení celku na jeho části.

Dodavatel (supplier). Osoba nebo osoby, které vyrábějí výrobek pro zákazníka. Ve smyslu tohoto návodu mohou být dodavatel i zákazník členy téže organizace.

Uživatel (user). Osoba nebo osoby, které přímo používají výrobek. Uživatel a zákazník často nejsou stejné osoby.

1 Požadavky na programové vybavení (PPV).

PPV je specifikace požadavků na část programu, celý program nebo soubor programů, který dělá určitou věc. Popis se skládá ze dvou základních požadavků na PPV:

1. Musí obsahovat určitá fakta. Například program, sestavený podle PPV, které neobsahují specifikaci chybových hlášení, nebude pravděpodobně vyhovovat zákazníkovi.
2. Tato fakta musí být vyjádřena určitým způsobem. Například program, sestavený podle PPV, ve kterých není specifikován formát ani obsah chybových hlášení a místo toho je uveden neurčitý požadavek „Všechna chybová hlášení budou prospěšná“, bude také nevyhovující pro zákazníka.

Je důležité uvědomit si roli, kterou hrají PPV v celém programovém projektu. Ustanovení standardu [2] definují minimální nezbytnou dokumentaci programového projektu. Poněvadž PPV má přesně určenou roli v tomto dokumentačním schématu, nesmí autor PPV v žádném případě překročit hranice této role. To znamená, že musí splnit následující požadavky:

1. PPV musí přesně definovat všechny požadavky na program, ale ne více.
2. PPV nesmí popisovat žádnou část návrhu, ověřování nebo řízení projektu, s výjimkou vyžadovaných omezení návrhu.

Vhodně sestavené PPV omezují sice oblast správných řešení, ale nspecifikují žádnou část návrhu a ponechávají dodavateli maximální volnost pro řešení. Správně sestavené PPV musí být:

1. jednoznačné (unambiguous)
2. úplné (complete)
3. ověřitelné (verifiable)
4. konsistentní (consistent)
5. modifikovatelné (modifiable)
6. sledovatelné (traceable)
7. použitelné ve fázi provozu a údržby.

1.1 Jednoznačnost.

PPV jsou jednoznačné tehdy a jen tehdy, když všechny v nich uvedené požadavky dovoluují pouze jediný výklad.

1. Minimálním požadavkem přitom je, že každá charakteristika výsledného produktu je popsána použitím jednoduchých a jedinečných pojmů.
2. V případě, že je v nějaké souvislosti použit termín, mající více významů, musí být zařazen do slovníku zvláštních výrazů, kde je použitý význam přesněji specifikován.

Požadavky jsou často napsány v přirozeném jazyce (například v angličtině). Autor PPV, který používá přirozený jazyk, musí zvláště pečlivě zkontrolovat jednoznačnost svých požadavků. Následující příklady jsou převzaty z 2. části [7].

1. Věta „Soubor dat bude obsahovat znak konce souboru“ může být čtena jako:
 - (a) Bude tam jeden a pouze jeden znak konce souboru.
 - (b) Některý znak bude určen jako znak konce souboru.
 - (c) Bude tam nejméně jeden znak konce souboru.
2. Věta „Celkový součet je vzat z posledního záznamu“ může být čtena jako:
 - (a) Celkový součet je vzat ze záznamu na konci souboru.
 - (b) Celkový součet je vzat z dosud posledního záznamu.
 - (c) Celkový součet je vzat z předcházejícího záznamu.
3. Věta „Všichni zákazníci mají stejné řídicí pole“ může být čtena jako:
 - (a) Všichni zákazníci mají stejnou hodnotu ve svém řídicím poli.
 - (b) Řídicí pole všech zákazníků mají stejný formát.
 - (c) Jedno řídicí pole je určeno pro všechny zákazníky.
4. Věta „Všechny soubory jsou řízeny pomocí bloku pro řízení souboru“ může být čtena jako:
 - (a) Jeden řídicí blok řídí celou množinu souborů.
 - (b) Každý soubor má svůj vlastní blok.
 - (c) Každý soubor je řízen řídicím blokem, ale jeden řídicí blok může řídit více než jeden soubor.

Jednou z možností jak zamezit nejednoznačností, které jsou vlastní přirozeným jazykům, je psaní PPV pomocí formálního jazyka pro specifikaci požadavků. Podrobnější popis tohoto tématu je uveden v [6], [13], [14] a [15].

1. Jednou z hlavních výhod použití takových jazyků je omezení víceznačnosti. To je částečně zajištěno tím, že formální jazykové procesory automaticky detekují řadu lexikálních, syntaktických i sémantických chyb.
2. Jednou z hlavních nevýhod použití těchto jazyků je dlouhý čas, nutný pro jejich naučení.

1.2 Úplnost.

PPV jsou úplné, jestliže mají následující vlastnosti:

1. Obsahují všechny významné požadavky, které se týkají funkčnosti, výkonnosti, omezení návrhu, vlastností nebo styku s vnějším prostředím.
2. Definují odezvy programu na všechny realizovatelné třídy vstupních dat při všech situacích, ke kterým může dojít. Poznamenejme, že je důležité stanovit odezvy na platné i neplatné vstupní hodnoty.
3. Dodržují všechny standardy pro PPV, které se jich týkají. Jestliže některá část standardu není zachována, PPV musí obsahovat číslo této části a vysvětlení, proč nebyla dodržena.
4. Všechny obrázky, tabulky a diagramy v PPV jsou označeny a lze se na ně odkazovat, stejně jako na definice všech pojmů a jednotek měření.

1.3 Ověřitelnost.

PPV jsou ověřitelné tehdy a jen tehdy, když je ověřitelný každý z požadavků, které jsou v něm uvedeny. Požadavek je ověřitelný tehdy a jen tehdy, když existuje nějaký konečný a cenově dostupný proces, kterým osoba nebo stroj mohou ověřit, že program zcela splňuje tento požadavek.

1. Příklady neověřitelných požadavků jsou obsaženy např. ve větách:
 - (a) „Program musí pracovat správně“ nebo „Program musí mít správné spojení s člověkem“. Tyto požadavky nemohou být ověřeny, protože nelze definovat pojmy „správný“ nebo „správně“.
 - (b) „Program se nesmí dostat do nekonečné smyčky“. Tento požadavek není ověřitelný, protože jeho otestování je teoreticky nemožné.
 - (c) „Výstup programu musí být obvykle dodán do 10 sekund“. Tento požadavek je neověřitelný, protože pojem „obvykle“ nemůže být změřen.
2. Příkladem ověřitelného výrazu je „Výstup programu musí být dodán do 20 sekund od události X po 60% času a do 30 sekund od události X po 100% času“. Tento požadavek může být ověřen, protože užívá přesné pojmy a měřitelné hodnoty.

3. Jestliže nemůže být vymyšlena metoda, která určí, zda program splňuje příslušný požadavek, pak musí být tento požadavek vypuštěn nebo revidován.
4. Jestliže nemůžeme vyjádřit požadavek pomocí ověřitelných pojmů v době přípravy PPV, musíme určit okamžik ve vývojovém cyklu projektu (oponentura, vydání testovacích dat), kdy musí být požadavky zadány v ověřitelné formě.

1.4 Konzistence.

PPV jsou konzistentní tehdy a jen tehdy, když žádná množina požadavků není ve vzájemném rozporu. Existují tři typy konfliktů konzistence v PPV:

1. Dva nebo více požadavků popisuje objekty téhož reálného světa, ale používá různé pojmy pro tytéž objekty. Například výzva programu na zadání vstupních dat může být v jednom požadavku označena „Zadej“ a v jiném „Ulož“.
2. Uvedené charakteristiky objektů reálného světa v různých požadavcích mohou být v rozporu. Například:
 - (a) Formát výstupu může být v jednom požadavku označen jako tabelární a v jiném jako textový.
 - (b) V jednom požadavku může být konstatováno, že barva znaku musí být zelená, zatímco v jiném je pro tentýž znak požadována modrá.
3. Může existovat logický nebo časový rozpor mezi dvěma uvedenými akcemi. Například:
 - (a) Jeden požadavek vyžaduje, aby dvě vstupní veličiny byly sečteny, zatímco jiný požaduje, aby je program vynásobil.
 - (b) Jeden požadavek tvrdí, že akce A musí vždy následovat za akcí B, zatímco jiný vyžaduje, aby A a B probíhaly současně.

1.5 Modifikovatelnost.

PPV jsou modifikovatelné, jestliže jejich struktura a způsob zápisu jsou takové, že všechny změny požadavků mohou být udělány snadno, úplně a konzistentně. Modifikovatelnost obecně vyžaduje, aby PPV:

1. Měly souvislou a snadno použitelnou organizaci s tabulkou obsahů, indexací a explicitními křížovými odkazy.
2. Nebyly redundantní, to jest aby týž požadavek nebyl uveden na více místech v PPV.

- (a) Redundance sama není chybou, ale může snadno k chybě vésti. Redundance může příležitostně pomoci udělat PPV čitelnější, problém však nastane při nutnosti modifikovat redundantní dokument. Předpokládejme například, že určitý požadavek stojí na dvou místech v PPV. Později je rozhodnuto, aby byl tento požadavek změněn, ale změna je provedena pouze na jednom z těchto míst. Tím se PPV stanou nekonsistentní.
- (b) Jestliže je redundance nezbytná, musí PPV obsahovat explicitní křížové odkazy, které zachovají jejich modifikovatelnost.

1.6 Sledovatelnost.

PPV jsou sledovatelné, jestliže je jasný původ každého požadavku a jestliže umožňuje sledování odkazů všech požadavků při vývoji nebo rozšíření dokumentace v budoucnosti. Rozeznáváme dva typy sledovatelnosti:

1. Zpětná sledovatelnost (to je k předcházejícímu vývojovému stupni) závisí na tom, zda jsou v každém požadavku explicitně uvedeny všechny odkazy na jeho zdroje v dřívějších dokumentech.
2. Dopředná sledovatelnost (to je k veškeré dokumentaci, vzniklé na základě PPV) závisí na tom, zda každý požadavek v PPV má své jedinečné jméno nebo referenční číslo.

Jestliže některý požadavek v PPV reprezentuje rozdělení nebo odvození jiného požadavku, musí být zajištěny obojí, zpětná i dopředná sledovatelnost. Zvláště důležitá je dopředná sledovatelnost PPV v okamžiku, kdy program vstupuje do fáze využívání a údržby. Jestliže jsou kterékoliv dokumenty modifikovány, musí být možné zjistit všechny požadavky, které tato modifikace postihne.

1.7 Použitelnost ve fázi využívání a údržby.

PPV musí obsahovat i požadavky na fázi využívání a údržby, včetně případného nahrazení programu.

1. Údržba je často prováděna osobou, která není nijak spojena s původním vývojem programu. Drobné změny mohou být skutečně pomoci dostatečně komentovaného kódu. V případě změn většího rozsahu je nutná dokumentace požadavků. Z toho vyplývají dvě potřeby:
 - (a) PPV musejí být modifikovatelné.
 - (b) PPV musejí obsahovat záznam o všech zvláštních opatřeních, která se týkají jednotlivých složek, jako např.:
 - i. Zvláštní důležitosti (například když selhání může ohrozit bezpečnost nebo způsobit velké finanční nebo společenské ztráty).

- ii. Vztahu k pouze přechodným potřebám (například podporovat výstup dat na obrazovku, která může být brzy přepsána).
 - iii. Jejich původu (například funkce X má být kopírována vcelku z existujícího programu).
2. Znalosti tohoto typu jsou dostupné v organizaci, která program vyvíjela, ale chybí v organizaci, která se zabývá údržbou. Jestliže účel nebo původ některé funkce nelze zjistit, je často nemožné zajistit její programovou údržbu.

2 Společná příprava PPV.

Proces vývoje programového vybavení začíná dohodou mezi dodavatelem a zákazníkem o tom, co úplný program musí dělat. Tato dohoda ve formě PPV musí být vypracována společně. To je důležité, protože zpravidla ani dodavatel ani zákazník není schopen připravit správné PPV sám.

1. Zákazník obvykle nerozumí návrhu programového vybavení a vývojovému postupu natolik, aby mohl napsat sám správné PPV.
2. Dodavatel zpravidla nezná dosti dobře zákaznickovy problémy a nemůže proto přesně sám specifikovat požadavky na vyhovující program.

Zákazník a dodavatel potřebují pracovat společně, aby vytvořili správně zapsané a úplně srozumitelné PPV. Tento návod neurčuje způsob diskuse, použitý jazyk ani techniky správného zápisu. Je jisté důležité, aby PPV byly správně zapsány; přehled o technice psaní návodů je uveden v [1] a [11]. Podle toho jak postupuje pokrok při vývoji programů, potřebují se rozvíjet i PPV.

1. Nebývá možné určit všechny podrobnosti projektu v okamžiku, když práce na projektu začínají. Například není možné definovat ve fázi zadávání požadavků detailní formáty všech obrazovek interaktivního programu způsobem, který zaručuje, že později nebudou upravovány.
2. Během vývoje programu se v PPV objevují nedostatky a nepřesnosti, z nichž vyplývají dodatečné změny. V tomto procesu jsou dvě hlavní hlediska:

3 Nástroje pro vývoj PPV.

Nejobvyklejším způsobem vytvoření PPV je jejich zápis pomocí přirozeného jazyka (například češtiny). Protože však přirozený jazyk je bohatý a nepřesný, byla vymyšlena řada formálnějších metod, které pomáhají autorům PPV. Jak dalece mohou být takové formální metodiky prospěšné při přípravě PPV záleží na řadě faktorů:

1. velikosti a složitosti programu,
2. zda to zákazník ve smlouvě požadoval,

3. zda PPV jsou prostředkem ke získání kontraktu nebo pouze vnitřním dokumentem,
4. zda se dokumenty s PPV stanou nejvyšší úrovní projektové dokumentace,
5. která počítačová zařízení máme k dispozici pro podporu této metodiky.

V tomto návodu není popisován žádný konkrétní nástroj. Podrobněji je tento námět diskutován např. v [6], [13], [14] nebo [15].

4 Požadavky na programové vybavení.

Požadavky mohou být vyjádřeny některým z následujících způsobů:

1. Pomocí vstupních a výstupních specifikací.
2. Pomocí souboru vzorových příkladů.
3. Pomocí specifikačních modelů.

Často je užitečné specifikovat požadované chování programu jako posloupnost vstupů a výstupů. Existují nejméně tři různé přístupy, založené na vlastnostech programu, který má být specifikován.

1. Některé programové produkty (například oznamovací systémy) je možné nejlépe popsat pomocí požadovaných výstupů. Obecně pracují výstupově orientované systémy především s datovými soubory. Vstup uživatele slouží zpravidla pouze k řízení informací a spouštění procesů nad datovými soubory.
2. Jiné systémy je možné nejlépe popsat, soustředíme-li se na chování vstupů a výstupů. Vstupně/výstupně orientované systémy zpracovávají zpravidla plynulý vstup dat. Slouží ke generování odpovídajících výstupů (např. pomocí podprogramů pro konverzi dat nebo souboru programů pro matematické funkce).
3. Od některých systémů (například pro řízení procesů) požadujeme, aby si pamatovaly svůj stav (své chování) tak, aby jejich odezva na nějaký vstup závisela i na předcházejících vstupech. Chovají se tedy podobně jako konečný automat. V tomto případě se musíme zaměřit na dvojice vstupů a výstupů a na posloupnosti těchto dvojic.

Je typické, že všechny požadavky, které se týkají programového vybavení, nejsou stejně důležité. Některé požadavky mohou být klíčové, zvláště pro životně důležité aplikace, zatímco jiné mohou být pouze „hezké, že existují“.

1. Každý požadavek v PPV musí obsahovat poznámku, která zajistí, aby vzájemné rozdíly v důležitosti jednotlivých požadavků byly jasné a zřetelné.
2. Poznámky k požadavkům, provedené tímto způsobem pomáhají:

- (a) Zákazníkům pečlivěji zvážit každý požadavek, což často vyjasní nějakou další skrytou možnost, kterou požadavek představuje.
- (b) Dodavatelům rozhodnout o správném způsobu návrhu a věnovat přiměřené úsilí různým částem programového produktu.

5 Rámcový prototyp PPV.

Tato kapitola pojednává o všech hlavních částech PPV. Jednotlivé části jsou uspořádány v tabulce 1. jako návrh, který může sloužit jako prototyp PPV. Dodavatelé programů a zákazníci by měli konkretizovat obsah požadavků podle tohoto návodu a jednotlivé společnosti na něm mohou založit své vlastní standardy pro PPV. Připomeňme, že i když nějaké PPV nebudou dodržovat tento návod nebo budou užívat rozdílné pojmy pro jeho části, musí každé správné PPV obsahovat všechny informace, které jsou zde popsány.

Tab.1. Rámcový prototyp PPV.

Obsah:

- 1. Úvod
 - 1.1. Účel
 - 1.2. Rozsah
 - 1.3. Definice, akronymy a zkratky
 - 1.4. Odkazy
 - 1.5. Přehled
- 2. Celkový popis
 - 2.1. Perspektiva výrobku
 - 2.2. Funkce výrobku
 - 2.3. Charakteristiky uživatele
 - 2.4. Obecná omezení
 - 2.5. Předpoklady a závislosti
- 3. Stanovené požadavky
 - 3.1. Funkční požadavky
 - 3.1.1. Funkční požadavek 1
 - 3.1.1.1. Úvod
 - 3.1.1.2. Vstupy
 - 3.1.1.3. Zpracování
 - 3.1.1.4. Výstupy
 - 3.1.1.5. Požadavky na výkon
 - 3.1.1.6. Omezení návrhu
 - 3.1.1.6.1. Shoda se standardy
 - 3.1.1.6.2. Omezení způsobená počítači
 - .
 - .
 - .
 - 3.1.1.7. Vlastnosti

- 3.1.1.7.1. Bezpečnost
- 3.1.1.7.2. Udržovatelnost
- .
- .
- .
- 3.1.1.8. Další požadavky
 - 3.1.1.8.1. Báze dat
 - 3.1.1.8.2. Postupy
 - 3.1.1.8.3. Přizpůsobení pracovišti
 - .
 - .
 - .
- 3.1.2. Funkční požadavek 2
- .
- .
- .
- 3.1.n. Funkční požadavek n
- 3.2. Požadavky na vnější rozhraní
 - 3.2.1. Uživatelské rozhraní
 - 3.2.1.1. Požadavky na výkon
 - 3.2.1.2. Omezení návrhu
 - 3.2.1.2.1. Shoda se standardy
 - 3.2.1.2.2. Omezení způsobená počítači
 - .
 - .
 - .
 - 3.2.1.3. Vlastnosti
 - 3.2.1.3.1. Bezpečnost
 - 3.2.1.3.2. Udržovatelnost
 - .
 - .
 - .
 - 3.2.1.4. Další požadavky
 - 3.2.1.4.1. Báze dat
 - 3.2.1.4.2. Postupy
 - 3.2.1.4.3. Přizpůsobení pracovišti
 - 3.2.2. Počítačové rozhraní
 - 3.2.3. Programové rozhraní
 - 3.2.4. Komunikační rozhraní

Dodatky

Index

6 Vysvětlivky k rámcovému prototypu

6.1 Úvod (Kapitola 1.)

Následující odstavce musí zajistit přehled o celých PPV.

Účel (Kapitola 1.1.)

Tento odstavec musí obsahovat následující:

1. Načrtnout účel vytváření konkrétních PPV.
2. Specifikovat, komu jsou PPV určeny.

Rozsah (Kapitola 1.2.)

Tento odstavec musí:

1. Jmenovitě určit o jaký programový produkt půjde; například hostitelský systém pro správu báze dat, generátor zpráv apod.
2. Vysvětlit, co programový produkt bude a pokud je to potřeba i co nebude dělat.
3. Popsat důležité vlastnosti specifikovaného programu. Tento popis musí:
 - (a) popsat všechny závažné vlastnosti a cíle tak přesně, jak je to jen možné. Například říci, že jedním cílem je dosáhnout „schopnosti účinně podávat hlášení“ není tak správné jako říci „parametry řízené a uživatelem definované zprávy s dvouhodinovou obrátkou a přímým vstupem uživatelských parametrů“.
 - (b) být v souladu s podobnými popisy na vyšší úrovni specifikace (například specifikace požadavků na systém), pokud existují.

Definice, akronymy a zkratky (Kapitola 1.3.)

Tento odstavec musí obsahovat definice všech pojmů, akronymů a zkratk ke správné interpretaci požadavků v PPV. Tyto informace mohou být opatřeny odkazem na jeden nebo více dodatků v PPV nebo v jiných dokumentech.

Odkazy (Kapitola 1.4.)

Tento odstavec musí:

1. Obsahovat úplný seznam všech dokumentů, na které jsou odkazy kdekoli v PPV nebo v oddělených souvisejících dokumentech.
2. Identifikovat každý dokument názvem, číslem odkazu a jestliže je to možné i datem a vydavatelem.
3. Uvést zdroj, ze kterého může být odkazovaný dokument získán. Tato informace může být zajištěna odkazem na dodatek nebo na jiný dokument.

Přehled (Kapitola 1.5.)

Tento odstavec musí:

1. popsat co je obsahem zbývající části PPV,
2. vysvětlit, jak jsou PPV organizovány.

6.2 Celkový popis (Kapitola 2.)

Tato kapitola PPV musí popisovat obecné skutečnosti, které mají vliv na výrobek a na požadavky na výrobek. Zpravidla obsahuje pět následujících subkapitol:

1. perspektiva výrobku
2. funkce výrobku
3. charakteristiky pro uživatele
4. obecná omezení
5. předpoklady a závislosti.

Musí být jasně řečeno, že tento odstavec neobsahuje vlastní požadavky, slouží pouze ke snadnějšímu porozumění těmto požadavkům.

Perspektiva výrobku (Kapitola 2.1.)

Tento odstavec provádí srovnání výrobku s jinými podobnými výrobky nebo projekty.

1. Jestliže výrobek je nezávislý a zcela samostatně použitelný, musí to být uvedeno zde.
2. Jestliže PPV definují výrobek, který je součástí nějakého většího systému nebo celku (jak se často stává), pak tento odstavec musí:
 - (a) Popsat funkci každé složky většího systému nebo projektu a identifikovat metody vzájemného styku.
 - (b) Identifikovat základní způsoby styku tohoto programového produktu s vnějším prostředím. Poznámka: Toto není podrobný popis těchto styků - ten je popsán jinde v PPV.
 - (c) Popsat hardware počítače a použité periferie.
Poznámka: Toto je pouze přehledný popis.

Blokový diagram, znázorňující hlavní složky většího systému, způsoby propojení a styk s vnějším prostředím může být velmi užitečný. Tento odstavec nesmí obsahovat přesné řešení návrhu ani přesná omezení návrhu nějakého řešení. V tomto odstavci je popsán pouze důvod, proč určitá omezení návrhu budou uvedena později jako část kapitoly Stanovené požadavky v PPV.

Funkce výrobku. (Kapitola 2.2.)

Tento odstavec PPV musí obsahovat přehled všech funkcí, které má program provádět. Například PPV pro program na vedení účetnictví může v tomto odstavci použít formulaci „vedení zákaznickova účtu, příkaz zákazníka, příprava faktur“ bez vysvětlování obrovského množství detailů, které by každá z těchto funkcí vyžadovala. Někdy může být přehled funkcí výrobku, potřebný pro tento odstavec přímo převzat z části, obsahující specifikace vyšší úrovně (pokud existuje), která obsahuje přesný popis funkcí programového produktu. Pro větší jasnost poznamenejme:

1. Funkce na seznamu musí být uvedeny takovým způsobem, že mu musí rozumět zákazník nebo kdokoliv jiný, kdo čte tento dokument poprvé.
2. Je užitečné zhotovit blokový diagram, ukazující různé funkce a jejich vzájemné vztahy. Připomeňme, že takový diagram není vyžadován při vlastním návrhu programu; je to jednoduchý a účinný nástroj pro vysvětlování.

Tento odstavec nesmí být použit k přesnému uvedení stanovených požadavků. Jsou zde pouze popsány důvody, proč určité stanovené požadavky budou uvedeny později, jako část kapitoly Stanovené požadavky v PPV.

Charakteristiky uživatele (Kapitola 2.3.)

Tento odstavec PPV musí obsahovat takové obecné charakteristiky eventuálního uživatele programového produktu, které budou mít vliv na stanovení požadavků. Po dobu života programového produktu, zvláště ve fázi využívání a údržby, s ním přijde do styku řada lidí. Jsou to uživatelé, operátoři, údržbáři, systémoví inženýři apod. Některé charakteristické rysy těchto lidí, jako úroveň vzdělání, zkušenost a technická zručnost, znamenají důležitá omezení pro zadání systémového okolí. Jestliže se systémem mohou často pracovat i příležitostní uživatelé, měl by být specifikován požadavek, aby systém obsahoval i návody, jak zadávat hlavní funkce. Nelze totiž předpokládat, že tito uživatelé si budou pamatovat různé detaily od minulé seance nebo že je přečtou v uživatelské příručce. Ani tento odstavec neobsahuje vlastní požadavky na programový produkt, ani neklade žádná omezení na návrh řešení. V tomto odstavci je popsán pouze důvod, proč určité specifické požadavky nebo omezení návrhu budou popsány později jako část kapitoly Stanovené požadavky v PPV.

Obecná omezení (Kapitola 2.4.)

Tento odstavec musí obsahovat obecný popis všeho, co by mohlo omezovat práci při vývoji a návrhu systému. Například může obsahovat:

1. způsob regulace
2. omezení hardware, například rychlosti odezvy na signál
3. způsob styku s jinými aplikacemi
4. paralelní operace

5. revizní funkce
6. řídicí funkce
7. požadavky na jazyky vyšších úrovní
8. protokoly výměny signálů, například XON - XOFF, ACK NACK.
9. kritické aplikace
10. úvahy o bezpečnosti a zabezpečení.

Ani tento odstavec neobsahuje vlastní požadavky na programový produkt, ani neklade žádná omezení na návrh řešení. V tomto odstavci je popsán pouze důvod, proč určité specifické požadavky nebo omezení návrhu budou popsány později jako část kapitoly Stanovené požadavky v PPV.

Předpoklady a závislosti (Kapitola 2.5.)

Tento odstavec PPV musí obsahovat seznam všech faktorů, které mohou hrát důležitou roli při stanovení požadavků v PPV. Tyto faktory samy neomezují návrh programu, ale jejich změny mohou mít vliv na stanovení požadavků v PPV. Například zde musí být uveden předpoklad, že na počítači, kde bude provozován specifikovaný programový produkt, bude k dispozici určitý operační systém. Jestliže ve skutečnosti tento operační systém k dispozici není, PPV musí být podle toho změněny.

6.3 Stanovené požadavky (Kapitola 3.)

Tato kapitola PPV musí obsahovat všechny podrobnosti, které autoři programu potřebují k vytvoření jeho návrhu. Je to zpravidla nejrozsáhlejší a nejdůležitější část PPV.

1. V této kapitole musejí být definovány jednotlivé stanovené požadavky při dodržení pokynů, které byly popsány v kapitole 3. tohoto návodu (ověřitelnost, jednoznačnost atd).
2. Všude, kde je to možné, musejí být vyznačeny křížové odkazy každého stanoveného požadavku na příslušné zmínky v úvodu, obecném popisu i dodatcích PPV.
3. Jeden způsob, jak klasifikovat stanovené požadavky je tento:
 - (a) požadavky na funkce
 - (b) požadavky na výkon
 - (c) omezení návrhu
 - (d) vlastnosti systému
 - (e) požadavky na vnější rozhraní.

Důležité vlastnosti, které by měly být známy jsou:

1. Stanovené požadavky musí být uspořádány logickým a čitelným způsobem.
2. Každý požadavek musí být stanoven tak, aby jeho dosažení mohlo být objektivně ověřeno předepsaným způsobem.

Požadavky na funkce.

Tento odstavec PPV musí obsahovat popis, jak musejí být vstupy do programového produktu transformovány na výstupy. Popisuje základní činnosti, které musí program provádět. Pro každou třídu funkcí nebo někdy pro každou samostatnou funkci je nutno specifikovat požadavky na vstupy, požadavky na zpracování a požadavky na výstupy. To je zpravidla zajištěno pomocí následujících čtyř odstavců:

1. **Úvod.** Tento odstavec PPV musí obsahovat popis důvodu zahrnutí funkce nebo funkcí a popis způsobu, jak budou zajištěny. Musí obsahovat všechny skutečnosti, které mohou přispět k vyjasnění účelu těchto funkcí.
2. **Vstupy.** Tento odstavec musí obsahovat:
 - (a) Podrobný popis všech datových vstupů pro tuto funkci, který zahrnuje:
 - i. zdroje vstupů
 - ii. počty
 - iii. jednotky měření
 - iv. časování
 - v. určení oblastí platných vstupů včetně přesnosti a tolerancí.
 - (b) Podrobné požadavky na činnost operátora, které musí obsahovat jméno a popis zásahu operátora a postavení ovládacího panelu nebo operátora. Například zde může být požadavek na operátorskou činnost při přerušení tisku - vložení a nastavení formuláře.
 - (c) Odkazy na dokumenty, ve kterých je specifikován styk s vnějším prostředím nebo řízení tohoto styku tam, kde je to nutné.
3. **Zpracování.** Tento odstavec musí definovat všechny operace, které musí být provedeny se vstupními daty a mezivýsledky tak, abychom získali výstupní data. Zahrnuje specifikaci:
 - (a) kontroly platnosti vstupních dat,
 - (b) přesnou posloupnost operací včetně časování všech událostí,
 - (c) odezvy na abnormální situace, například:
 - i. přetečení
 - ii. chybu při komunikaci
 - iii. zpracování chyby,
 - (d) parametrů, které jsou nezbytné pro prováděné operace,

- (e) požadavků pro znehodnocení operací,
- (f) všech metod (například rovnice, matematické algoritmy a logické operace), které mají být použity při transformaci vstupů na odpovídající výstupy. Například zde mohou být specifikovány:
 - i. vzorce pro výpočet daní v programu pro výplaty
 - ii. metoda nejmenších čtverců v programu pro kreslení křivek
 - iii. použitý meteorologický model v programu pro předpovídání počasí,
- (g) kontroly platnosti výstupních dat.

4. **Výstupy.** Tento odstavec musí obsahovat:

- (a) Přesný popis všech datových výstupů této funkce, který se skládá z:
 - i. místa určení výstupu
 - ii. množství
 - iii. jednotek měření
 - iv. časování
 - v. určení oblasti platných výstupů včetně přesnosti a tolerancí
 - vi. vyrovnání se s nepřipustnými hodnotami
 - vii. popisu chybových hlášení.
- (b) odkazy na dokumenty, ve kterých je specifikováno rozhraní s vnějším prostředím nebo řízení tohoto rozhraní tam, kde je to nutné.

Navíc u systémů, jejichž činnost je zvláště zeměřena na chování vstupů a výstupů musí být v PPV specifikovány dvojice vstupů a výstupů a posloupnosti těchto dvojic. Posloupnosti jsou důležité u systémů, u nichž je vyžadováno zapamatování předchozího chování tak, aby jejich odezva mohla záviset na vstupu i na stavu systému, to jest chovají se jako konečné automaty.

Požadavky na výkon.

Můžeme je rozdělit na statické a dynamické.

1. Statické numerické požadavky mohou obsahovat:

- (a) počet podporovaných terminálů,
- (b) počet uživatelů, kteří mohou být obslouženi současně,
- (c) počet souborů nebo záznamů s nimiž může být manipulováno,
- (d) rozměry tabulek a souborů.

Statické numerické požadavky jsou někdy zpracovány jako samostatný odstavec, nazvaný „Kapacita“.

2. Dynamické numerické požadavky mohou obsahovat například počet transakcí, počet úloh, množství dat, která mohou být zpracována během nějakého časového úseku při běžném zatížení i při špičkovém zatížení.

Všechny tyto požadavky musí být uvedeny v měřitelných pojmech, například „95% transakcí musí být zpracováno dříve než za 1 sekundu“ a ne formou „operátor nesmí čekat na ukončení transakce“. Poznámka: Numerická omezení pro jednu funkci jsou zpravidla uvedena jako součást odstavce, ve kterém je popsáno zpracování této funkce.

Omezení návrhu.

Omezení návrhu mohou být způsobena použitím některých standardů, mezemi použitých počítačů apod.

Shoda se standardy.

Tento odstavec musí obsahovat požadavky, vyplývající z dodržování existujících norem a standardů. Může zde být například:

1. formát zpráv,
2. názvy dat,
3. výpočetní postupy,
4. sledování postupu revize, například zde může být uveden požadavek na zaznamenání všech aktivit při zpracování. Takové záznamy jsou v některých aplikacích vyžadovány pro splnění finančních nebo vládních norem. V požadavku na sledování celého postupu může být například uvedeno, že všechny změny ve výplatní datové bázi musí být zaznamenány ve zvláštním souboru, kde je vždy uvedena stará i nová hodnota.

Omezení způsobená počítači.

Tento odstavec by měl obsahovat takové požadavky na programové vybavení, které zajistí jeho použitelnost i na různých počítačích s různými omezeními. Například zde mohou být uvedeny:

1. charakteristické rysy počítačové konfigurace (počet portů, soubor instrukcí apod.)
2. meze pro použití vnitřní a vnější paměti.

Požadavky na vnější rozhraní.

Uživatelské rozhraní.

Zde musí být specifikovány:

1. Charakteristiky, které musí program zajistit pro umožnění spolupráce s člověkem. Jestliže například uživatel komunikuje s programem prostřednictvím obrazovkového terminálu, musí být zadáno následující:
 - (a) požadované formáty obrazovky,
 - (b) rozmístění na stránce a obsah všech zpráv nebo menu,
 - (c) vzájemné časování vstupů a výstupů,
 - (d) dostupnost některých funkcí pomocí klíčů.
2. Všechna hlediska, která mohou optimalizovat styk s člověkem, který obsluhuje program. To může jednoduše obsahovat seznam toho, co systém sdělí a nesdělí uživateli. Jedním příkladem může být požadavek na možnost volby mezi krátkými a dlouhými chybovými zprávami. Podobně jako všechny ostatní, i tyto požadavky musí být ověřitelné. Například formulace „písařka ve třídě 4 může provést funkci X po jedné hodině tréninku za Z minut“ je lepší než „písařka může provést funkci X “. (To může být uvedeno také v odstavci Vlastnosti jako část, nazvaná „Snadnost použití“.)

Počítačové rozhraní.

Zde musejí být uvedeny logické charakteristiky všech rozhraní mezi programovým produktem a počítačovými složkami systému. To také pokrývá popis toho, která zařízení mají být podporována, jak jsou podporována a použité protokoly. Například podpora terminálu může být specifikována jako podpora obrazovkového terminálu jako protiklad k řádkovému terminálu.

Programové rozhraní.

Zde musí být uvedeno použití všech jiných požadovaných programových produktů (například operační systém, systém pro správu dat, soubor matematických podprogramů) a rozhraní s jinými aplikačními systémy (například spojení mezi systémem pro účtování a obecným kontrolním systémem). Pro každý uvedený programový produkt musí být uvedeno následující:

1. jméno
2. mnemotechnika
3. číslo specifikace
4. číslo verze
5. pramen.

Pro každé rozhraní tato část musí:

1. Vysvětlit účel připojovaného programu ve vztahu k vlastnímu programovému produktu.

2. Definovat rozhraní pomocí obsahu přenášených zpráv a jejich formátu. Není nutné podrobně popisovat některé dříve dokumentované rozhraní, ale musí být uveden odkaz na dokument, kde je definice rozhraní popsána.

Komunikační rozhraní.

Zde musejí být uvedena různá rozhraní, sloužící ke komunikaci, jako například protokol lokální sítě apod.

Uspořádání stanovených požadavků.

Tato kapitola je často nejrozsáhlejší a nejobsažnější ze všech částí PPV.

1. Může být vhodné rozdělit tuto kapitolu na odstavce, podle tříd funkcí, které mají být programem prováděny. Představme si například rozsáhlý interaktivní účtovací systém. Ten může být na vrchní úrovni rozdělen na výkonné programy (které provádějí transakce téměř v reálném čase), podpůrné programy (zahájení seance, zálohování disků, zaplňování pásek apod.) a diagnostické programy (podporující zpravidla hardware a komunikace). Na další úrovni může být rozdělen na přijaté faktury, zaplacené faktury atd.
2. Nesmíme zapomenout, že hlavním důvodem tohoto způsobu rozdělení je zlepšení čitelnosti PPV a ne definování nejvyšší úrovně návrhu programového vybavení.

Nejlepší uspořádání části 3. Stanovené požadavky v PPV závisí na oblasti aplikace a na charakteristických rysech programového produktu, který má být specifikován.

Přídavné informace.

Přídavné informace, t.j. obsah, dodatky, index apod. činí PPV snadněji použitelnými.

1. Obsah a index jsou velmi důležité a musejí v nich být dodržována obecně platná pravidla pro zpracování právné dokumentace. Viz například [4] a [11].
2. Dodatky nejsou povinnou částí PPV a nemusejí být vždy uvedeny. Mohou obsahovat:
 - (a) Vzory vstupních a výstupních formátů, popisy studie cenové analýzy nebo výsledky uživatelských odhadů.
 - (b) Základní nebo podpůrné informace, které mohou pomoci čtenáři PPV.
 - (c) Popis problému, který má být řešen programovým vybavením.
 - (d) Historii, základní data, zkušenosti a charakteristické rysy organizace, pro kterou má být program vytvářen.
 - (e) Seznam křížových odkazů v němž jsou uvedeny dříve zhotovené nekompletní požadavky na programové vybavení. Viz [4], odstavce 4.3.2. a 4.3.3.
 - (f) Speciální instrukce na kódování a nosiče informací, zajišťující bezpečnost, exportovatelnost, počáteční zavedení programu nebo jiné požadavky.

3. Pokud PPV obsahuje dodatky, musí zde být explicitně uvedeno, zda tyto dodatky jsou nebo nejsou součástí požadavků.

Literatura

- [1] ANSI/IEEE Std 100–1977, IEEE Standard Dictionary of Electrical and Electronics Terms.
- [2] ANSI/IEEE Std 730–1981, IEEE Standard for Software Quality Assurance Plans.
- [3] ANSI/IEEE Std 729–1983, IEEE Standard Glossary of Software Engineering Terminology.
- [4] Brusaw, C.T. and Alred, G. and Oliu, W.: Handbook of Technical Writing, New York, St. Martin's Press, 1976.
- [5] Dasarathny, B.: Timing constraints of Real-Time Systems: Constructs for Expressing Them, IEEE Real-Time Systems Symposium, Dec. 1982.
- [6] Davis, A.: The Design of a Family of Applications-Oriented Requirements Languages, IEEE Computer, 15, 5 May 1982, pp 21–28.
- [7] Freedman, D. and Weinberg, G.: Handbook of Walkthroughs, Inspections and Technical Reviews, 3rd Ed. Little and Brown Publishers, New York.
- [8] Kain, R.: Automata Theory: Machines and Languages, McGraw Hill, New York, 1972.
- [9] Kohavi, Z.: Switching and Finite Automata Theory, McGraw Hill, New York, 1970.
- [10] Kramer, J.(ed): Application Oriented Specifications Glossary of Terms, European Workshop on Industrial Computer Systems (EWICS), Imperial College, London, England, May 6, 1981.
- [11] Mills, G. and Walter, J.: Technical Writing, New York, Holt, Rinehart and Winston, 5th Ed, 1978.
- [12] Peterson J.: Petri Nets, ACM Computing Surveys, 9, 4, Dec 1977, pp 223–252.
- [13] Ramamoorthy, C. and So, H.H.: Software Requirements and Specifications: Status and Perspectives, Tutorial: Software Methodology, IEEE Catalog No EHO 142-0, 1978, pp 43–164.
- [14] Taggart, W.M.Jr. and Tharp, M.O.: A Survey of Information Requirements Analysis Techniques, ACM Computing Surveys, 9, 4, Dec 1977, pp 273–290.

- [15] Teichroew, D.: A Survey of Languages for Stating Requirements for Computer-Based Information Systems, 1972 Fall Joint Computer Conference, 1972, pp 1203–1224.