Kapitola 2 – Softvérové procesy

Softvérový proces

- Štruktúrovaný súbor činností potrebných na vývoj softvérového systému.
- Existuje mnoho rôznych softvérových procesov, ale všetky zahŕňajú:
  - Špecifikácia – definovanie toho, čo má systém robiť;
  - Návrh a implementácia – definovanie organizácie systému a implementácia systému;
  - Validácia – kontrola, či robí to, čo zákazník chce;
  - Evolúcia – zmena systému v reakcii na meniace sa potreby zákazníkov.
- Softvérový procesný model je abstraktná reprezentácia procesu. Predstavuje popis procesu z určitej konkrétnej perspektívy.

Popisy softvérových procesov

- Keď popisujeme a diskutujeme procesy, zvyčajne hovoríme o činnostiach v týchto procesoch, ako je špecifikácia dátového modelu, návrh používateľského rozhrania atď. a poradie týchto činností.
- Opis procesov môže tiež zahŕňať:
  - Produkty, ktoré sú výsledkom procesnej činnosti;
  - Roly, ktoré odrážajú zodpovednosť ľudí zapojených do procesu;
  - Predbežné a následné podmienky- čo sú tvrdenia, ktoré sú pravdivé pred a po uzákonení procesnej činnosti alebo po vyrobení produktu.

Plánom riadené a agilné procesy

- Procesy riadené plánom ("plánované procesy") sú procesy, pri ktorých sú všetky procesné činnosti vopred naplánované a pokrok sa meria podľa tohto plánu.
- V agilných procesoch je plánovanie prírastkové a je jednoduchšie zmeniť proces tak, aby odrážal meniace sa požiadavky zákazníkov.
- V praxi väčšina procesov zahŕňa prvky plánom riadeného aj agilného prístupu.
- Neexistujú žiadne správne alebo nesprávne softvérové procesy.

Modely softvérových procesov

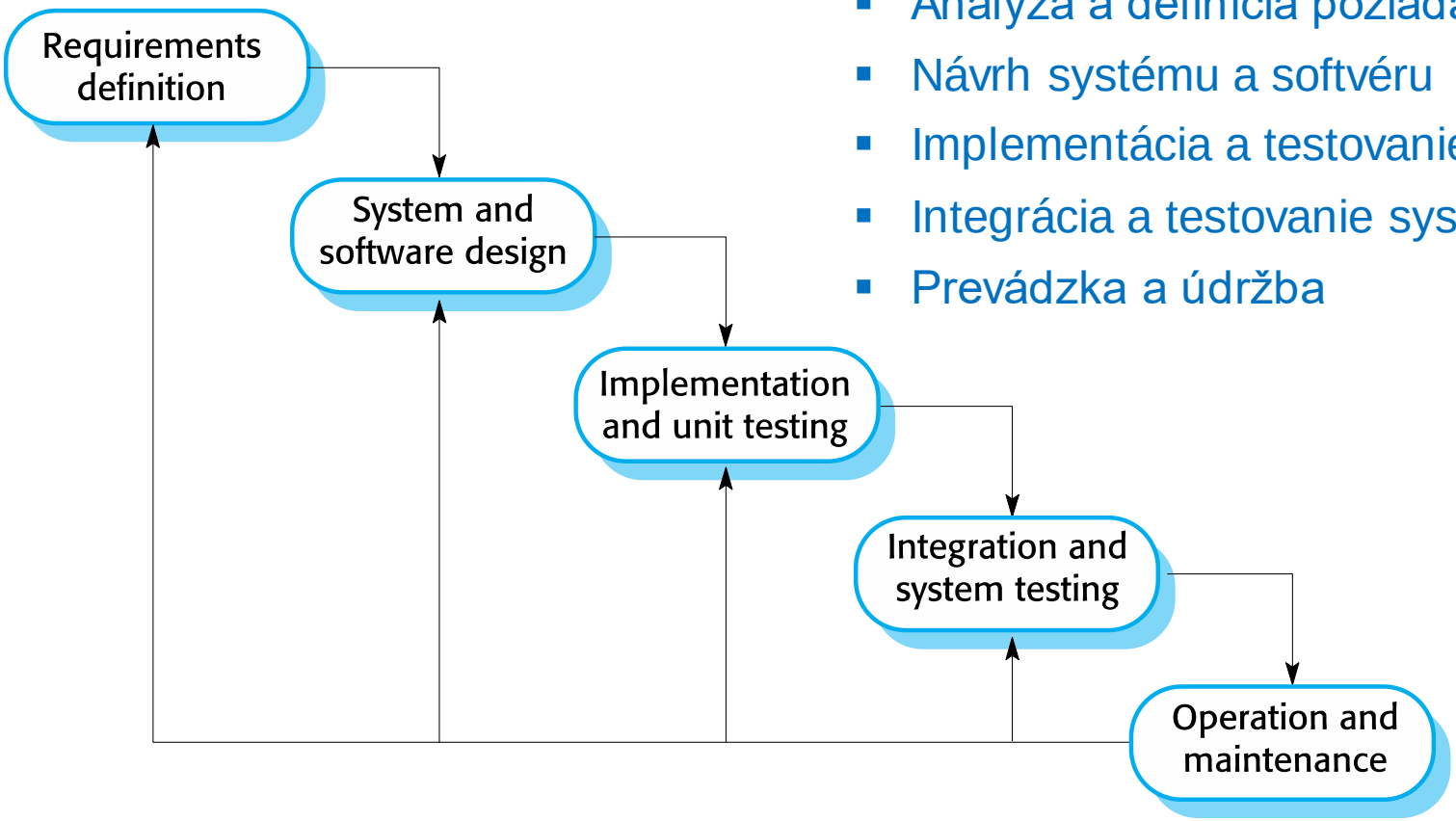
Softvérové procesné modely

- ✧ Vodopádový model
  - Model riadený plánom. Oddelené a odlišné fázy špecifikácie a vývoja.
- ✧ Postupný vývoj
  - Špecifikácia, vývoj a validácia sú navzájom prepojené. Môže byť riadený plánom alebo agilný.
- ✧ Integrácia a konfigurácia
  - Systém je zostavený z existujúcich konfigurovateľných komponentov. Môže byť riadený plánom alebo agilný.
- ✧ V praxi sa väčšina veľkých systémov vyvíja pomocou procesu, ktorý zahŕňa prvky zo všetkých týchto modelov.

(1) Vodopádový model

Fázy

- Analýza a definícia požiadaviek
- Návrh systému a softvéru
- Implementácia a testovanie jednotiek
- Integrácia a testovanie systému
- Prevádzka a údržba



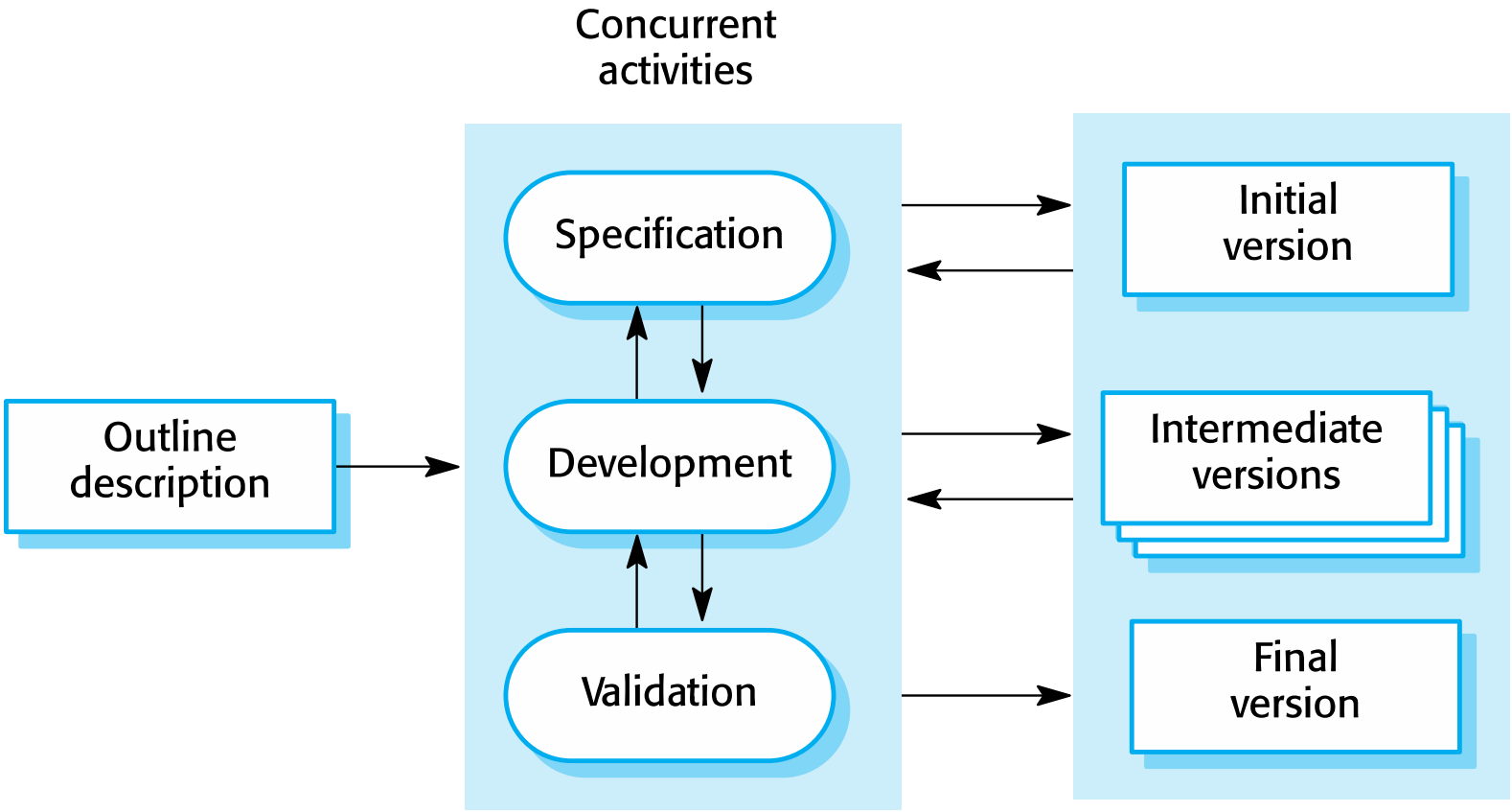
Fázy modelu vodopádu

- Vo vodopádovom modeli sú oddelené identifikované fázy:
  - Analýza a definícia požiadaviek
  - Návrh systému a softvéru
  - Implementácia a testovanie jednotiek
  - Integrácia a testovanie systému
  - Prevádzka a údržba
- Hlavnou nevýhodou vodopádového modelu je obtiažnosť prispôsobiť sa zmenám po tom, čo proces prebieha. V zásade musí byť fáza dokončená pred prechodom na ďalšiu fázu.

Problémy s modelom vodopádu

- Neflexibilné rozdelenie projektu na jednotlivé etapy sťažuje reakciu na meniace sa požiadavky zákazníkov.
  - Preto je tento model vhodný len vtedy, keď sú požiadavky dobre pochopené a zmeny budú počas procesu návrhu značne obmedzené.
  - Len málo obchodných systémov má stabilné požiadavky.
- Vodopádový model sa väčšinou používa pre veľké projekty systémového inžinierstva, kde sa systém vyvíja na niekoľkých miestach.
  - Za týchto okolností pomáha plánom riadený charakter modelu vodopádu koordinovať prácu.

(2) Postupný vývoj



Výhody postupného vývoja

- Znižujú sa náklady na prispôsobenie sa meniacim sa požiadavkám zákazníkov.
  - Množstvo analýz a dokumentácie, ktoré je potrebné prepracovať, je oveľa menšie, ako sa vyžaduje pri modeli vodopádu.
- Je jednoduchšie získať spätnú väzbu od zákazníkov na vykonanú vývojovú prácu.
  - Zákazníci môžu komentovať ukážky softvéru a vidieť, koľko sa implementovalo.
- Je možné rýchlejšie dodanie a nasadenie užitočného softvéru zákazníkovi.
  - Zákazníci môžu softvér používať a získavať z neho hodnotu skôr, ako je to možné pri vodopádovom procese.

Problémy s postupným vývojom

- Proces nie je viditeľný.
  - Manažéri potrebujú pravidelné výstupy na meranie pokroku. Ak sa systémy vyvíjajú rýchlo, nie je nákladovo efektívne vytvárať dokumenty, ktoré odrážajú každú verziu systému.
- Štruktúra systému má tendenciu degradovať, keď sa pridávajú nové prírastky.
  - Pokiaľ nie je vynaložený čas a peniaze na refaktorovanie na zlepšenie softvéru, pravidelné zmeny majú tendenciu poškodiť jeho štruktúru. Začlenenie ďalších softvérových zmien je čoraz ťažšie a nákladnejšie.

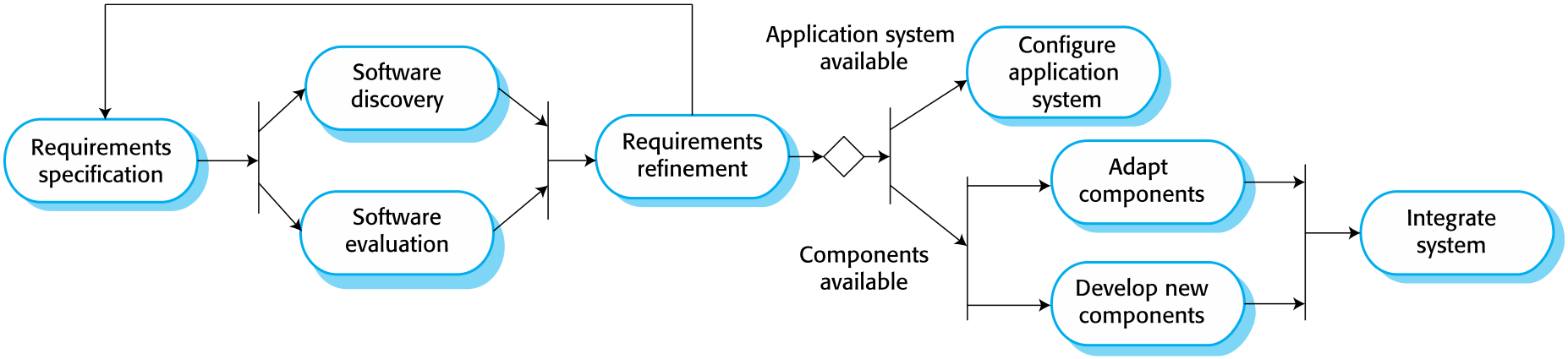
(3) Integrácia a konfigurácia

- Založené na opätovnom použití softvéru, kde sú systémy integrované z existujúcich komponentov alebo aplikačných systémov (niekedy nazývaných COTS – Commercial-off-the-shelf) systémov).
- Opätovne použité prvky môžu byť nakonfigurované tak, aby sa ich správanie a funkčnosť prispôsobili požiadavkám používateľa
- Opätovné použitie je v súčasnosti štandardným prístupom k budovaniu mnohých typov podnikových systémov
  - Opätovné použitie podrobnejšie popísané v kapitole 15, prednáška 9.

Typy opätovne použiteľného softvéru

- Samostatné aplikačné systémy (niekedy nazývané COTS), ktoré sú nakonfigurované na použitie v konkrétnom prostredí.
- Kolekcie objektov, ktoré sú vyvinuté ako balík na integráciu s komponentovým rámcom, ako je .NET alebo J2EE.
- Webové služby, ktoré sú vyvinuté podľa servisných štandardov a ktoré sú dostupné pre vzdialené vyvolanie.

Softvérové inžinierstvo orientované na opätovné použitie



Kľúčové fázy procesu

- Špecifikácia požiadaviek
- Objav a vyhodnotenie nájdeného softvéru
- Spresnenie požiadaviek
- Konfigurácia aplikačného systému
- Prispôsobenie a integrácia komponentov

Výhody a nevýhody

- Znížené náklady a riziká, pretože menej softvéru sa vyvíja od nuly
- Rýchlejšie dodanie a nasadenie systému
- Kompromisy požiadaviek sú však nevyhnutné, takže systém nemusí spĺňať skutočné potreby používateľov
- Strata kontroly nad vývojom opätovne použitých prvkov systému

Procesné činnosti

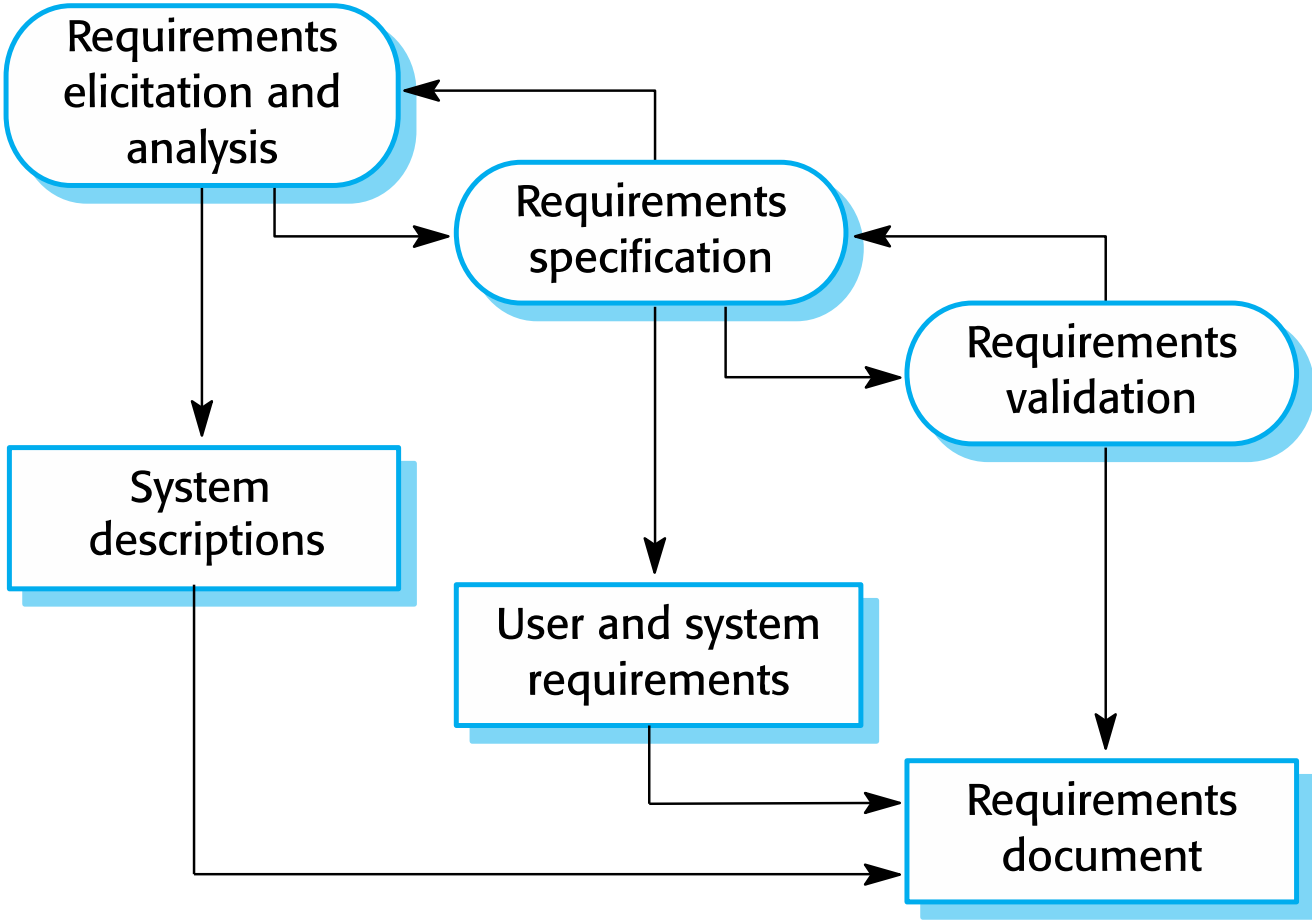
Procesné činnosti

- Reálne softvérové procesy sú vzájomne prepojené sekvencie technických, kolaboratívnych a manažérskych činností s celkovým cieľom špecifikovať, navrhnúť, implementovať a otestovať softvérový systém.
- Štyri základné procesné činnosti ("aktivity softvérového procesu") sú špecifikácie, vývoja, validácie a evolúcie a sú v rôznych vývojových procesoch organizované odlišne.
- Napríklad vo vodopádovom modeli sú usporiadané v poradí, zatiaľ čo v prírastkovom vývoji sú vykonávané súčasne.

(1) Špecifikácia softvéru

- Proces zisťovania, aké služby sú požadované a obmedzenia fungovania a rozvoja systému.
- Požiadavky na inžiniersky proces (viď obr.)
  - Získavanie a analýza požiadaviek
    - Čo od systému požadujú alebo očakávajú účastníci systému?
  - Špecifikácia požiadaviek
    - Detailné definovanie požiadaviek
  - Overenie požiadaviek
    - Kontrola platnosti požiadaviek

Analýza s špecifikácia požiadaviek



(2) Návrh a implementácia softvéru

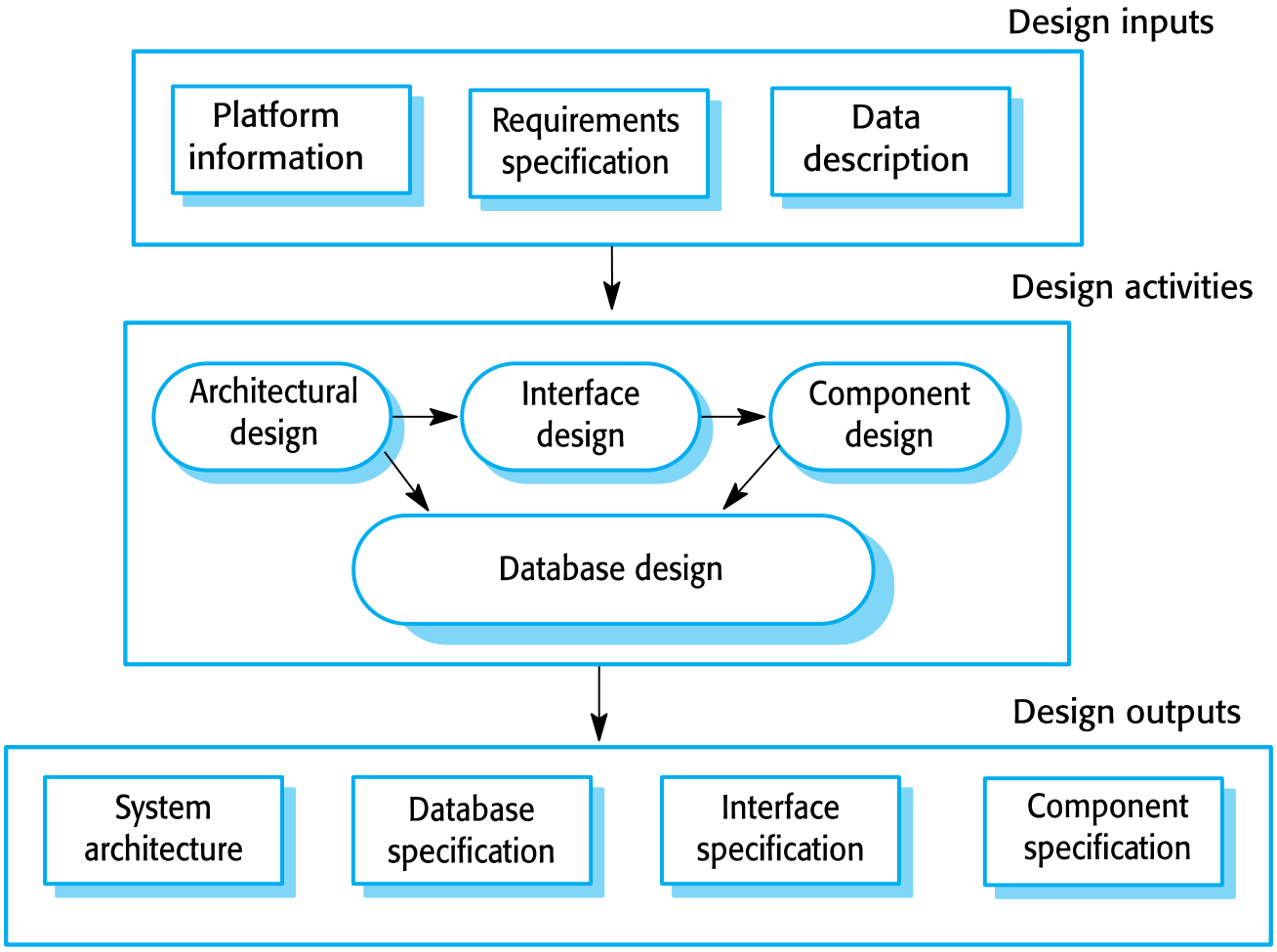
- Proces prevodu špecifikácie systému na spustiteľný systém.
- Návrh softvéru
  - Navrhnite softvérovú štruktúru, ktorá realizuje špecifikáciu;
- Implementácia
  - Preložte túto štruktúru do spustiteľného programu;

- Činnosti návrhu a implementácie spolu úzko súvisia a môžu sa navzájom dopĺňať.

(2a) Dizajnérske činnosti

- *Architektonický návrh*, kde identifikujete celkovú štruktúru systému, hlavné komponenty (subsystémy alebo moduly), ich vzťahy a spôsob ich rozloženia.
- *Návrh databázy*, kde navrhujete štruktúry údajov systému a ako majú byť reprezentované v databáze.
- *Návrh rozhrania*, kde definujete rozhrania medzi komponentmi systému.
- *Výber a dizajn komponentov*, kde opakovane hľadáte použiteľné komponenty. Ak nie sú k dispozícii, navrhnite, ako systém bude fungovať.

Všeobecný model procesu navrhovania



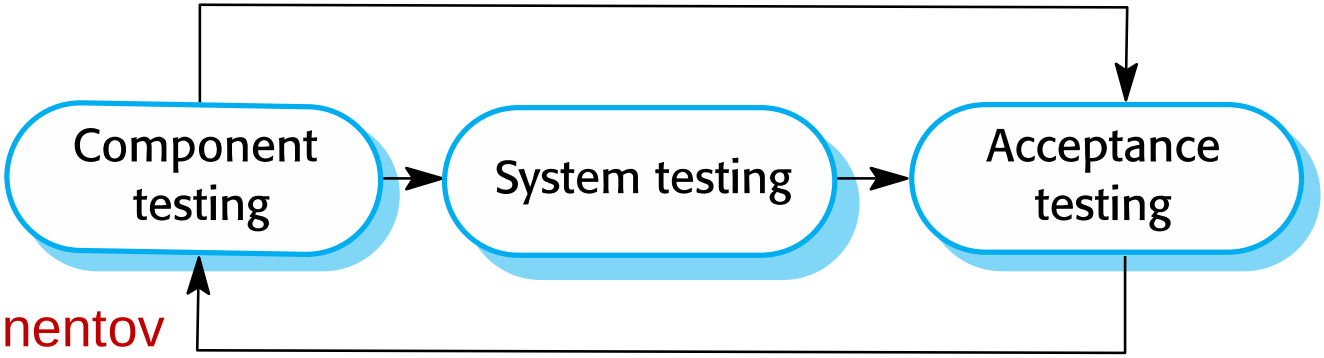
(2b) Implementácia systému

- Softvér sa implementuje buď vývojom programu alebo programov alebo konfiguráciou aplikačného systému.
- Návrh a implementácia sú prepojené činnosti pre väčšinu typov softvérových systémov.
- Programovanie je individuálna činnosť bez štandardného procesu.
- Ladenie je činnosť zameraná na vyhľadávanie chýb programu a nápravu týchto chýb.

(3) Validácia softvéru

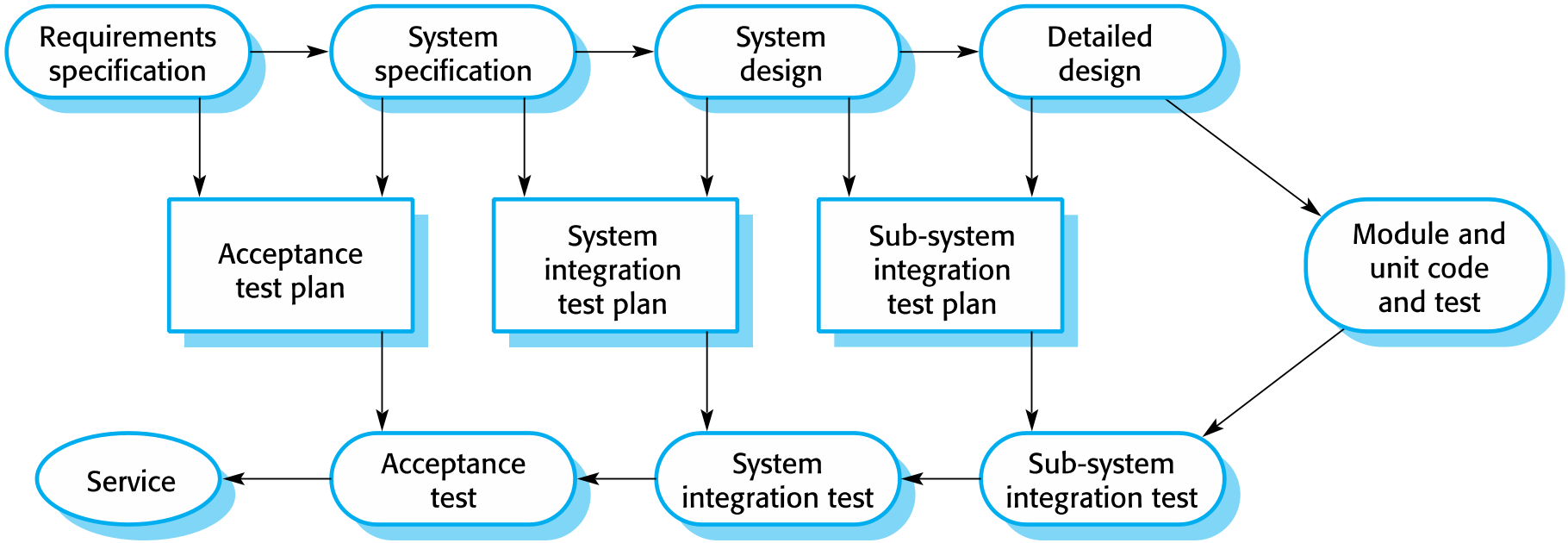
- Verifikácia a validácia (V & V) je určená na preukázanie, že systém zodpovedá svojej špecifikácii (verifikácia) a spĺňa požiadavky zákazníka systému (validácia).
- Zahŕňa procesy kontroly, preskúmania a testovanie systému.
- Testovanie systému zahŕňa spustenie systému pomocou testovacích prípadov, ktoré sú odvodené zo špecifikácie skutočných údajov, ktoré má systém spracovať.
- Testovanie je najčastejšie používaná V & V aktivita.

Fázy testovania



- Testovanie komponentov
  - Jednotlivé komponenty sú testované nezávisle;
  - Komponenty môžu byť funkcie alebo objekty alebo koherentné zoskupenia týchto entít.
- Testovanie systému
  - Testovanie systému ako celku. Zvlášť dôležité je testovanie emergentných vlastností.
- Zákaznícke testovanie
  - Testovanie s údajmi od zákazníkov na kontrolu, či systém spĺňa potreby zákazníka.

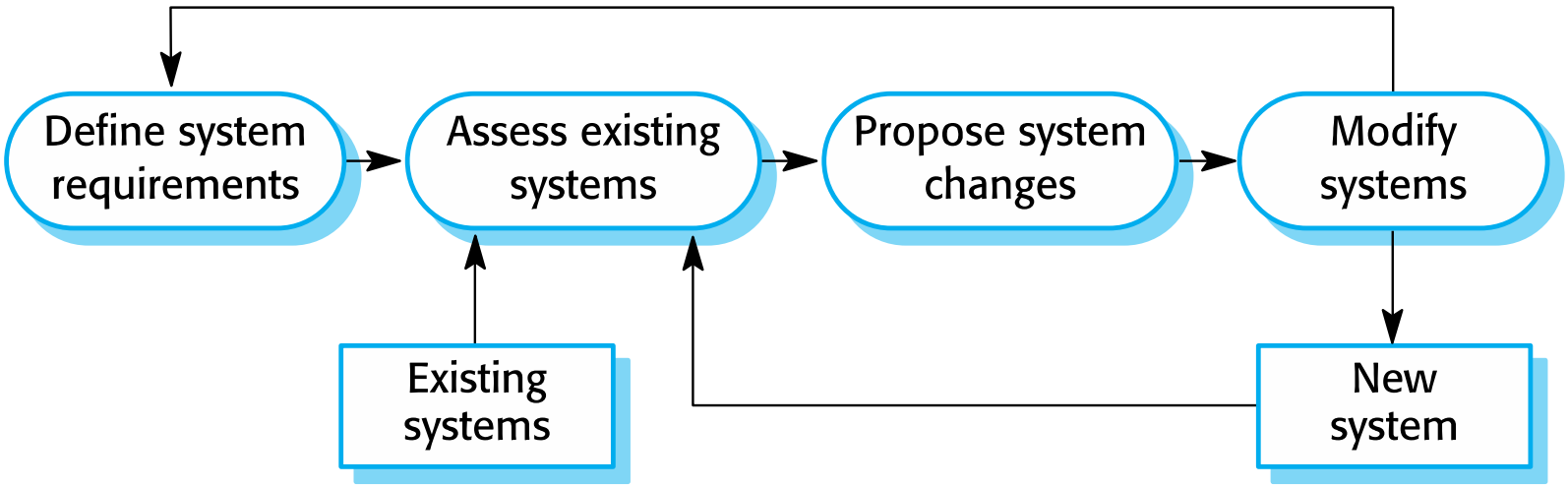
Fázy testovania v softvérovom procese riadenom plánom (V-model)



(4) Evolúcia alebo ďalší vývoj softvéru

- Softvér je vo svojej podstate flexibilný a môže sa meniť.
- Keďže požiadavky sa menia v dôsledku meniacich sa obchodných podmienok, musí sa vyvíjať a meniť aj softvér, ktorý podporuje podnikanie.
- Hoci existuje hranica medzi vývojom a evolúciou (údržbou), je to stále viac irelevantné, keďže stále menej systémov je úplne nových.

Evolúcia systému



Vyrovnanie sa so zmenou

Vyrovnanie sa so zmenou

- Zmena je nevyhnutná vo všetkých veľkých softvérových projektoch.
  - Obchodné zmeny vedú k novým a zmeneným systémovým požiadavkám
  - Nové technológie otvárajú nové možnosti na zlepšenie implementácií
  - Meniace sa platformy si vyžadujú zmeny aplikácií
- Zmena vedie k prepracovaniu, takže náklady na zmenu zahŕňajú prepracovanie (napr. prehodnotenie požiadaviek), ako aj náklady na implementáciu novej funkcionality.

Zníženie nákladov na prepracovanie

- Predvídanie zmien , kde softvérový proces zahŕňa činnosti, ktoré môžu predvídať možné zmeny predtým, ako je potrebné vykonať významné prepracovanie.
  - Napríklad môže byť vyvinutý prototyp systému, ktorý zákazníkom ukáže niektoré kľúčové vlastnosti systému.
- Tolerancia zmien , kde je proces navrhnutý tak, aby sa zmeny mohli prispôsobiť relatívne nízkym nákladom.
  - To zvyčajne zahŕňa určitú formu postupného vývoja. Navrhované zmeny môžu byť implementované v prírastkoch, ktoré ešte neboli vyvinuté. Ak to nie je možné, môže byť zmenený iba jeden prírastok (malá časť systému), aby sa začlenila zmena.

Vyrovnanie sa s meniacimi sa požiadavkami

- Systémové prototypovanie , kde sa rýchlo vyvinie verzia systému alebo časť systému, aby sa preverili požiadavky zákazníka a uskutočniteľnosť návrhových rozhodnutí. Tento prístup podporuje predvídanie zmien.
- Inkrementálne doručovanie , kde sa systémové inkrementy doručujú zákazníkovi na pripomienkovanie a experimentovanie. To podporuje vyhýbanie sa zmenám a toleranciu zmien.

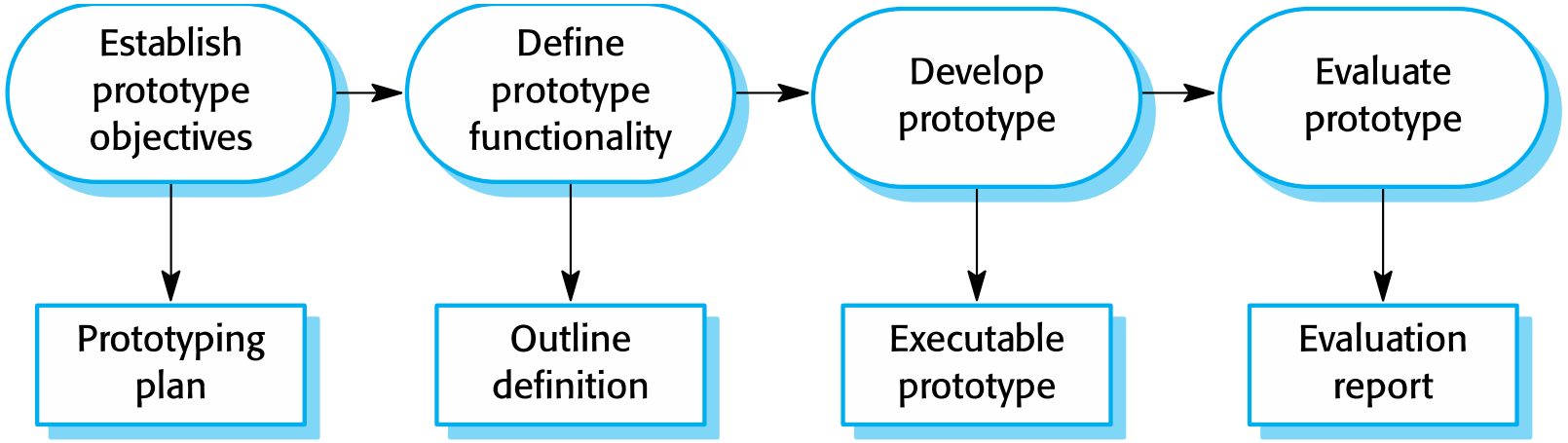
Prototypovanie softvéru

- Prototyp je počiatočná verzia systému používaná na demonštráciu konceptov a vyskúšanie možností dizajnu.
- Prototyp je možné použiť:
  - Proces analýzy požiadaviek na pomoc pri vytváraní a overovaní požiadaviek;
  - V procese návrhu preskúmať možnosti a vyvinúť dizajn používateľského rozhrania;
  - V procese testovania na spustenie testov

Výhody prototypovania

- Vylepšená použiteľnosť systému.
- Presnejšie prispôsobenie skutočným potrebám používateľov.
- Vylepšená kvalita dizajnu.
- Vylepšená udržiavateľnosť.
- Znížené úsilie o vývoj.

Proces vývoja prototypu



Vývoj prototypu

- Môžu byť založené na jazykoch alebo nástrojoch rýchleho prototypovania
- Môže zahŕňať vynechanie funkcií ("nevýhody")
  - Prototyp by sa mal zamerať na oblasti produktu, ktoré nie sú dobre pochopené;
  - Prototyp nemusí obsahovať kontrolu a obnovu chýb;
  - Zamerajte sa skôr na funkčné ako nefunkčné požiadavky, ako je spoľahlivosť a bezpečnosť

Vyhadzovacie prototypy

- Prototypy by sa mali po vývoji vyradiť, pretože nie sú dobrým základom pre produkčný systém:
  - Môže byť nemožné vyladiť systém tak, aby spĺňal nefunkčné požiadavky;
  - Prototypy sú zvyčajne nezdokumentované;
  - Štruktúra prototypu je zvyčajne degradovaná rýchlou zmenou;
  - Prototyp pravdepodobne nebude spĺňať bežné organizačné štandardy kvality.

"nevýhody"

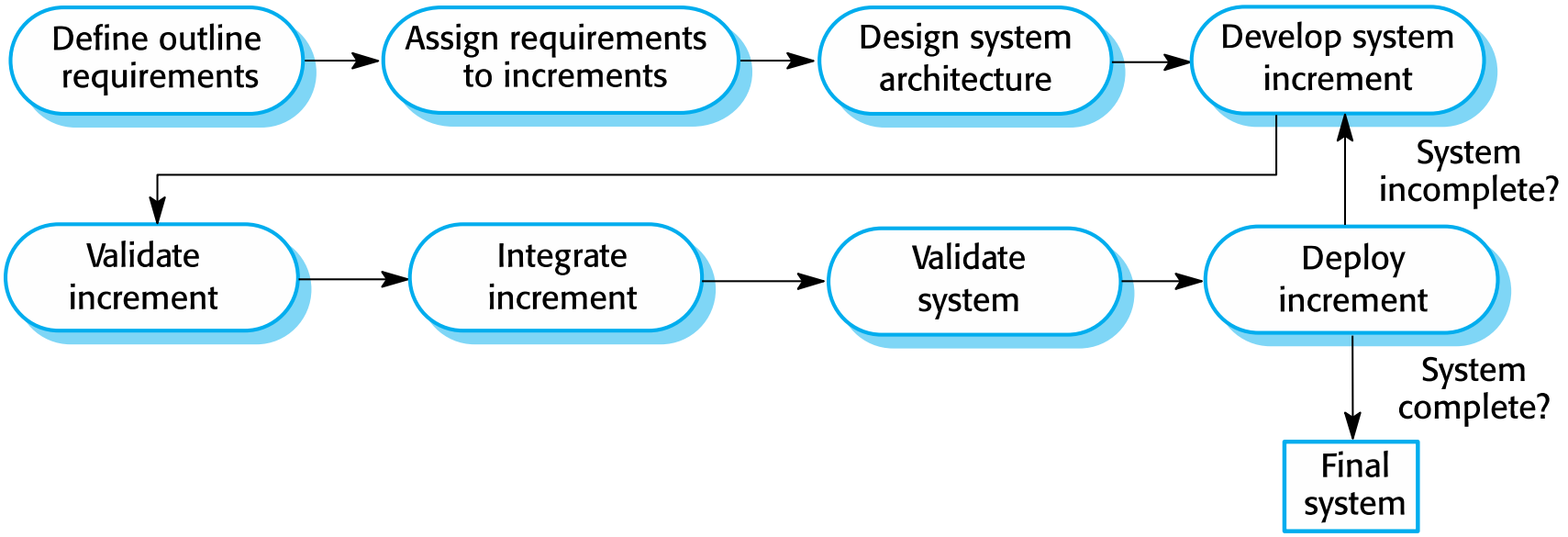
Prírastkové doručenie

- Namiesto dodania systému ako jednej dodávky je vývoj a dodávka rozčlenená na prírastky, pričom každý prírastok poskytuje časť požadovanej funkčnosti.
- Požiadavky používateľov sú uprednostňované a požiadavky s najvyššou prioritou sú zahrnuté v prvých prírastkoch.
- Po spustení vývoja prírastku sa požiadavky zmrazia, hoci požiadavky na neskoršie prírastky sa môžu naďalej vyvíjať.

Postupný vývoj a duručovanie

- Postupný vývoj
  - Vyvíjajte systém v prírastkoch a vyhodnoťte každý prírastok predtým, ako pristúpite k vývoju ďalšieho prírastku;
  - Normálny prístup používaný v agilných metódach;
  - Hodnotenie vykonáva používateľ/zástupca zákazníka.
- Prírastkové doručenie
  - Nasaďte prírastok, aby bol používaný koncovými používateľmi;
  - Realistickejšie hodnotenie praktického používania softvéru;
  - Je ťažké toto realizovať pri nahrádzaní systému, pretože prírastky majú menšiu funkčnosť ako vymieňaný systém.

Prírastkové doručenie



Výhody prírastkového doručovania

- Hodnota pre zákazníka môže byť poskytovaná s každým prírastkom, takže funkčnosť systému je k dispozícii skôr.
- Skoré prírastky fungujú ako prototyp, ktorý pomáha získať požiadavky na neskoršie prírastky.
- Nižšie riziko celkového zlyhania projektu.
- Systémové služby s najvyššou prioritou sú zvyčajne najviac testované.

Problémy s postupným doručovaním

- Väčšina systémov vyžaduje "základy", ktoré používajú rôzne časti systému.
  - Keďže požiadavky nie sú detailne definované, kým sa nezačne implementovať prírastok, môže byť ťažké identifikovať spoločné služby, ktoré sú potrebné pre všetky prírastky.
- Podstatou iteračných procesov je, že špecifikácia sa vytvára spolu so softvérom.
  - To je však v rozpore s modelom obstarávania mnohých organizácií, kde je kompletná špecifikácia systému súčasťou zmluvy o vývoji systému.

Vylepšenie procesov

Vylepšenie procesu

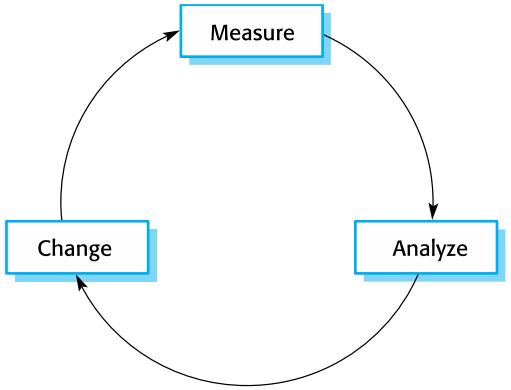
- Mnoho softvérových spoločností sa obrátilo na zlepšovanie softvérových procesov ako na spôsob zvyšovania kvality svojho softvéru, znižovania nákladov alebo urýchlenia procesov vývoja.
- Zlepšenie procesov znamená pochopenie existujúcich procesov a zmenu týchto procesov s cieľom zvýšiť kvalitu produktu a/alebo znížiť náklady a čas vývoja.

Prístupy k zlepšeniu

- Prístup procesnej zrelosti, ktorý sa zameriava na zlepšovanie procesného a projektového riadenia a zavádzanie dobrej praxe softvérového inžinierstva.
  - Úroveň zrelosti procesov odráža mieru, do akej bola v procesoch vývoja softvéru organizácie prijatá dobrá technická a manažérska prax.
- Agilný prístup, ktorý sa zameriava na iteratívny vývoj a znižovanie režijných nákladov v softvérovom procese.
  - Primárnou charakteristikou agilných metód je rýchle poskytovanie funkčnosti a schopnosť reagovať na meniace sa požiadavky zákazníkov.

Činnosti na zlepšenie procesov

- *Procesné meranie*
  - Meriate jeden alebo viac atribútov softvérového procesu alebo produktu. Tieto merania tvoria základ, ktorý vám pomôže rozhodnúť, či boli zlepšenia procesov efektívne.



- *Procesná analýza*
  - Súčasný proces sa hodnotí a identifikujú sa slabé miesta a úzke miesta procesu. Môžu byť vyvinuté modely procesov (niekedy nazývané mapy procesov), ktoré popisujú proces .
- *Zmena*
  - procesné zmeny na odstránenie niektorých zistených slabých stránok procesov. Tieto sa zavedú a cyklus sa obnoví na zber údajov o účinnosti zmien .

Procesné meranie

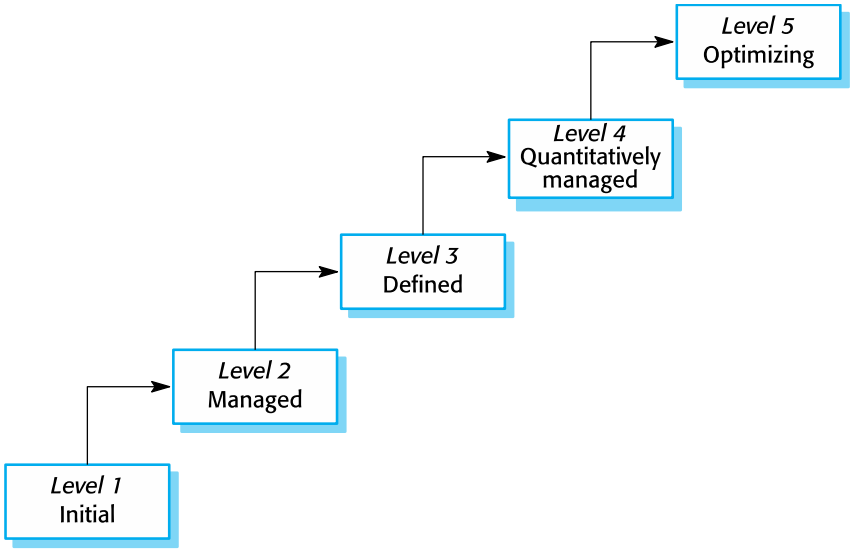
- Mali by sa zbierať kvantitatívne údaje o procese
  - Ak však organizácie nemajú jasne definované procesné štandardy, je to veľmi ťažké, pretože neviete, čo merať. Pred možným meraním môže byť potrebné definovať proces.

- Procesné merania by sa mali používať na posúdenie zlepšenia procesov
  - To však neznamená, že merania by mali viesť k zlepšeniam. Motorom zlepšovania by mali byť organizačné ciele.

Procesné metriky

- Čas potrebný na dokončenie procesných činností
  - Napr. čas alebo úsilie v kalendári na dokončenie činnosti alebo procesu.
- Zdroje potrebné na procesy alebo činnosti
  - Napr. celkové úsilie v osobo-dňoch.
- Počet výskytov konkrétnej udalosti
  - Napr. počet zistených defektov.

Procesy - úrovne zrelosti spôsobilosti



5. Optimalizovené - Stratégie zlepšovania procesov definované a používané

4. Organizované - Definované a používané stratégie manažérstva kvality

3. Definované a používané postupy a stratégie procesného riadenia

2. Opakovateľné - Definované a používané postupy riadenia produktu

1. Počiatočné - V podstate nekontrolované