

MONITOROVANIE SOFTVÉ- ROVÉHO SYSTÉMU ANALÝZOU VYTVORENEJ HODNOTY

Dôveruj, ale preveruj. Plánuj, ale monitoruj.

Andrej Kozák

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
akozak86[zavináč]gmail[.]com

Abstrakt. *Analýza vytvorenej hodnoty (EVA) je metóda projektového manažmentu pre kontrolu priebehu projektu. Dva problémy však spôsobujú ťažkosti v aplikovaní tejto metódy na softvérové projekty. Prvým je otázka, ako dôsledne vyjadriť základné línie softvérového projektu a merať jeho technický vývoj. Druhý problém je zapríčinený štruktúrou delenia práce (WBS), ktorá napomáha rozdelením práce na elementy lepšej organizácii a kontrole vývoja, no nemôže zachytiť a odraziť prípadné softvérové zmeny vo vývoji. Táto esej najskôr približuje prístupy používané pri monitorovaní projektov – analýzu vytvorenej hodnoty, štruktúru rozdelenia práce a metódu bodov prípadov použitia (UCP). Následne sa zaoberá problematikou a riešením použitia analýzy vytvorenej hodnoty pri softvérových projektoch, prezentuje evolučnú štruktúru delenia práce zameranú na jednotlivé elementy softvérových procesov a opisuje prevzatie prípadov použitia (UCP) ako základných softvérových línií projektu, merania efektivity a výkonu. Práca v závere hodnotí prínos analýzy vytvorenej hodnoty v spojení s metódou bodov prípadov použitia do monitorovania vývoja softvérových projektov..*

Kľúčové slová: *softvérový projekt, monitorovanie a kontrola, analýza vytvorenej hodnoty, body prípadov použitia, štruktúra rozdelenia práce*

Monitorovanie

Softvérové projekty aj napriek tomu, že sú mnohokrát dobre a precízne naplánované, sa nevyvíjajú podľa plánu. Spravidla platí, že čím väčší projekt, tým k väčším odchýlkam dochádza počas jeho životného cyklu. Podľa prieskumov sa 70% projektov stretá s dvoma najväčšími problémami:

1. prekročený rozpočet
2. nedodržaný termín ukončenia

Ten istý prieskum uvádza, že 52% všetkých projektov končí na 189% ich plánovaného rozpočtu, čo je takmer dvojnásobok.[4] No a niektoré z nich ani po veľkých investíciách a vynaloženom úsilí nie sú napokon dokončené. Príčin takýchto nežiaducich stavov je mnoho a ani dôsledným plánovaním sa im mnohokrát nemožno vyhnúť. Okolitý svet totiž neustále podlieha zmenám, ktoré sa postupne prejavujú aj na projektoch, no nebolo ich možné v prvých fázach projektu zapracovať do plánu, podľa ktorého sa teraz postupuje.

Monitorovanie projektov je práve tá činnosť, ktorá má tieto zmeny priebežne odhaľovať a zohľadňovať pri tvorbe ďalších plánov, prípadne prijať opatrenia na minimalizáciu negatívneho vplyvu na projekt. Monitorovaním projektu sledujeme určité jeho vlastnosti a priebeh, záleží to od jednotlivého spôsobu monitorovania. Treba si však položiť otázku: Ktoré parametre/atribúty softvérového projektu sledovať, aby sme vedeli určiť, nakoľko postupujeme podľa plánu? Pre túto esej som zvolil metódu analýzy vytvorenej hodnoty v spojení s metódou bodov prípadov použitia.

Analýza vytvorenej hodnoty

Analýza vytvorenej hodnoty je medzinárodne uznávaná technika projektového manažmentu, ktorá je široko používaná na kontrolu stavu projektu a predvídanie ďalšieho pokračovania vývoja či realizácie.

Analýza vytvorenej hodnoty berie do úvahy tri základné parametre:

- Plánovaná hodnota – plánované náklady na úlohy, ktoré mali byť vykonané za určité časové obdobie. Táto položka odpovedá na otázku, koľko plánujeme investovať do projektu a aká práca na projekte má byť ukončená do daného termínu
- Vytvorená hodnota – predstavuje hodnotu práce, ktorá bola skutočne dokončená.
- Aktuálne náklady – náklady na ukončenie práce v zadanom termíne. V skutočnosti zodpovedá nákladom, ktoré boli minuté v procese vytvárania hodnoty

Najdôležitejšími indikátormi stavu projektu sú práve

1. odchýlka nákladov = vytvorená hodnota - aktuálne náklady
2. odchýlka plánu = vytvorená hodnota - plánovaná hodnota

Záporná hodnota pri odchýlke nákladov znamená, že projekt prekročil rozpočet určený pre jeho vývoj, nepriaznivá odchýlka plánu sa objaví, ak aktuálne náklady na vytvorenú

hodnotu prekročia rozpočet vyhradený pre celú prácu. Priebeh projektu je stanovený porovnaním vytvorenej hodnoty k projektovej línii v priebehu celého vývoja projektu. Všetky tieto hodnoty sa značia do jedného diagramu pre jednoduchú ilustráciu.

Analýza vytvorenej hodnoty teda ponúka spôsoby, ako merať vykonanú prácu na projekte, predpovedať ďalšie náklady, odhadnúť čas dokončenia, určiť ako projekt napreduje v porovnaní s pôvodným plánom a vychádzajúc z týchto poznatkov predpovedať, ako sa projekt bude vyvíjať v budúcnosti.

Pri tejto analýze sa vyžaduje, aby bola všetka práca na projekte rozdelená na úlohy alebo aktivity, kvôli optimálnemu plánovaniu, procesu vývoja a kontrole. Vytvorená hodnota potom musí byť meraná zvlášť pre každú úlohu s prihliadnutím na hodnotu skutočných výsledkov. Práve to nás privádza k problému odhadov v softvérových projektoch, plánovania a merania jednotlivých úloh alebo technického vývoja v softvérových projektoch[3].

Štruktúra rozdelenia práce

V projekte monitorovanom analýzou vytvorenej hodnoty je všetka práca rozdelená do tzv. pracovných balíčkov, čo sú akoby menšie podúlohy, ktoré spolu súvisia. Tieto pracovné balíčky sú organizované do stromovej štruktúry a každý z nich má tri hlavné charakteristiky: technická stránka, termín (časový plán) a rozpočet. Na obrázku 1 je príklad rozčlenenia práce na pracovné balíčky na rôznych úrovniach.



Obr. 1. Štruktúra rozdelenia práce ilustrovaná na procese prípravy pizze.

Štruktúra rozdelenia práce je obvyčajne organizovaná subsystémami a komponentmi, ktoré formujú systémové riešenie. Napríklad štruktúra rozdelenia práce orientovaná na produkt vyžaduje, aby bol dizajn softvéru daný pred začiatkom projektu, čo v praxi nemôže byť vždy splnené. Okrem iného, preddefinovaná štruktúra softvéru a jeho komponentov nie je schopná odraziť skutočné napredovanie vývoja softvérového projektu.

V oblasti softvérového vývoja boli realizované výskumy zamerané na softvérové procesy. Štruktúra rozdelenia práce orientovaná nie na dizajn systému, ale na proces, sa môže vyvíjať nezávisle a bez narušenia systémovej štruktúry. Táto štruktúra zároveň poskytuje lepšie základy pre definovanie projektových línii[2]. Ako ale definujeme projektové línie pri softvérovom projekte? Jedným z možných riešení je použitie metódy bodov prípadov použitia.

Body prípadov použitia

Modely pre predpovedanie vývoja softvérového projektu a pre meranie práce si nie sú adekvátne. Bežné metriky, ako počet riadkov kódu sú neúplné a zavádzajúce, pretože vývoj softvéru zahŕňa oveľa viac ako písanie kódu, napr. návrh architektúry, revízia dizajnu, testovanie a podobne. No a tieto činnosti zaberať obyčajne väčšie množstvo času v softvérovom projekte ako samotná implementácia, teda písanie kódu.

Na prvé odhady dĺžky vývoja softvérového projektu ešte v prvých fázach, keď už sú presne zadefinované požiadavky na funkcionality, môže byť použitá analýza funkčných bodov. Táto metóda je rozvinutá do bodov prípadov použitia ako objektovo orientovaná technológia a prípady použitia sú široko praktizované.[1]

Prístup bodov prípadov použitia bol navrhnutý na odhad vývoja softvérových projektov už v skorších fázach. Táto metóda je meraná z modelu prípadov použitia, ktorý definuje funkčný rozsah systému, ktorý sa vyvíja. Dva hlavné prvky, z ktorých sa ráta bod prípadu použitia, sú diagram prípadov použitia na systémovej úrovni a tok udalostí. Prvý z nich zahŕňa jeden alebo viac diagramov prípadov použitia, ktoré definujú všetky prípady použitia a všetkých hráčov v systéme. Druhý, teda tok udalostí, zahŕňa sekcie pre normálne a alternatívne cesty (spojenia) v každom prípade použitia. Bod prípadov použitia je potom meraný zrátaním počtu hráčov a transakcií zahrnutých v toku udalostí s nejakou váhou. Transakcia je v tejto metóde udalosť, ktorá nastane medzi hráčom a cieľovým systémom. Odhad úsilia pomocou bodu prípadov použitia je vykonávaný v šiestich krokoch:

1. zrátanie váhy hráčov – hráči v modeloch prípadov použitia sú rozdelení na jednoduchých (systém s aplikačným programovým rozhraním), priemerných (iný systém komunikujúci cez protokol, osoba komunikujúca cez terminál) a komplexných (osoba interagujúca cez grafické používateľské rozhranie). Počet jednotlivých typov hráčov je zrátaný a vynásobený tabuľkovou hodnotou priradenou jednotlivým stupňom úrovne hráčov. Váha hráčov sa potom vypočíta ako suma týchto hodnôt.
2. zrátanie váhy prípadov použitia – podobne ako pri hráčoch, aj prípady použitia sú rozdelené do na jednoduché, priemerné a komplexné, no tu sa berie do úvahy počet transakcií prípadu použitia. Následne sa spočítajú počty prípadov použitia v jednotlivých skupinách a tie sú vynásobené tabuľkovou hodnotou (váhovým faktorom) pre jednotlivé skupiny. Nakoniec sa váhy zrátajú a dostávame váhu prípadov použitia
3. vyrátanie neupravených bodov prípadov použitia – neupravené body prípadov použitia dostaneme jednoduchým pripočítaním váhy hráčov (z bodu 1) k váhe prípadov použitia (bod 2)
4. zváženie technických faktorov a environmentálnych faktorov – v tomto bode sa vypočíta technický faktor a faktor prostredia. Treba pritom čo najlepšie analyzovať všetky možné faktory vplývajúce na dané procesy a priradiť im rozsah vplyvu – číslo od 0 do 5, kde 0 znamená, že faktor je irelevantný pre projekt, 5 znamená zásadný vplyv faktoru na projekt. V tabuľke 1 a 2 sú možné technické a environmentálne faktory vplývajúce na projekt. Z týchto tabuľkových

údajov (faktory a ich váhy) sa násobia hodnoty faktorov, ktoré sme si určili a vypočítajú sa 2 hodnoty: technický faktor (TCF) a environmentálny faktor (EF)

5. vypočítanie bodov prípadov použitia - tieto body sú vypočítané podľa formuly: neupravené body prípadov použitia * technický faktor * faktor prostredia
6. odhad úsilia – odhad úsilia (práce), ktoré treba vynaložiť na ukončenie projektu získame vynásobením hodnoty bodov prípadov použitia a špecifickej hodnoty. Podľa výskumov Schneidera a Wintersa [6] je táto hodnota navrhnutá na 20 hodín na bod prípadu použitia.

Tab. 1. Technické faktory a ich váhy pre systém

Faktor	Opis	Váha
TF1	Distribučný systém	2
TF2	Komplexné vnútorné spracovanie	1
TF3	Kód musí byť znovupoužiteľný	1
TF4	Jednoduchá inštalácia	0,5
TF5	Jednoduché používanie	0,5
TF6	Prenosný	1
TF7	Jednoduchá zmena	1
TF8	Súbežnosť	1
TF9	Zahŕňa špeciálne bezpečnostné prvky	1
TF10	Priamy prístup pre tretie strany	1

Tab. 2. Environmentálne faktory a ich váhy pre používateľa

Faktor	Opis	Váha
EF1	Skúsenosti s aplikáciou	0,5
EF2	Skúsenosti s objektovo orientovaným programovaním	1
EF3	Analytické schopnosti	0,5
EF4	Motivácia	1
EF5	Stabilné požiadavky	2
EF6	Pracovníci na polovičný úväzok	-1
EF7	Zložitý programovací jazyk	-1

Máme teda rozdelenú prácu a po šiestich krokoch sme vyrátali aj odhad úsilia, ktoré treba vynaložiť pre jednotlivé prípady použitia a pre úspešné ukončenie projektu. Nasleduje teda analýza zozbieraných údajov a jej využitie pri monitorovaní softvérového projektu.

Analýza vytvorenej hodnoty založená na bodoch prípadov použitia

Presnosť odhadov projektu závisí od úrovne abstrakcie, ktorá je známa v čase odhadu. Platí teda, že čím presnejšie informácie (špecifikáciu) k projektu máme, tým lepšie vieme odhadnúť potrebnú prácu a náklady na projekt. Avšak problémom softvérových projektov je práve to, že nie všetky detaily sú známe v začiatkoch projektu. Na riešenie tohto problému bola navrhnutá tzv. vývojová štruktúra rozdelenia práce.

V prvom rade sa klasifikuje práca na projekte do dvoch druhov podľa toho, do akej miery o práci vieme, že má byť ukončená. Podľa toho balíčky štruktúry rozdelenia práce spadajú pod pracovné balíčky (na ktorých sa bude pracovať) a plánované balíčky (ktoré sú len plánované ale práca ešte nezačne).

Potom práca s blízkym termínom ukončenia je rozdelená do pracovných balíčkov, čo sú časti práce, ktoré možno naplánovať do detailu. Pracovné balíčky slúžia ako projektové línie v čase, keď sú presne zadefinované a sú pripravené na vykonávanie.

Práca s neskorším termínom ukončenia je pridelená do plánovaných balíčkov, ktoré majú slabú úroveň detailov. V priebehu realizácie projektu, ako sa získavajú detaily a informácie k práci zaradenej v plánovacích balíčkoch, tieto balíčky postupne prechádzajú do skupiny pracovných balíčkov. Tým následne získava projekt nové línie.

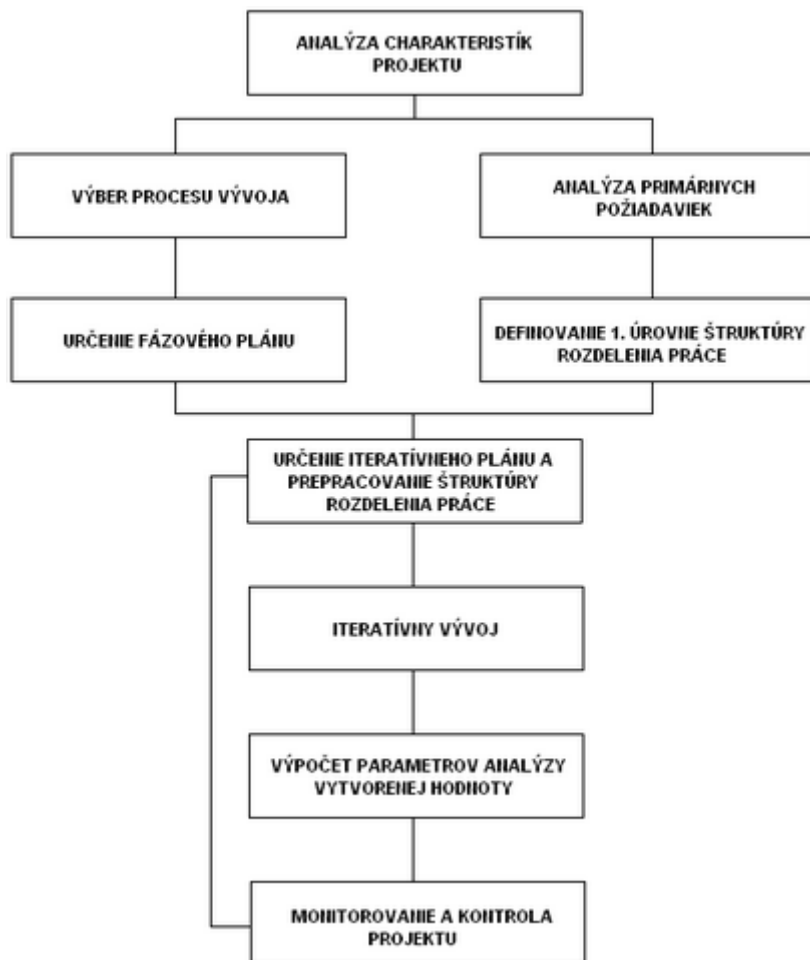
Následne po rozdelení práce do balíčkov, štruktúra rozdelenia práce organizuje elementy v rámci procesu a nie samotného riešenia produktu. Práve preto sa táto štruktúra lepšie adaptuje na zmeny počas vývoja projektu. Keďže systém je vyvíjaný iteratívne a inkrementálne, štruktúra rozdelenia práce sa zdokonaľuje a dynamicky prispôsobuje a rozširuje s ďalšími dostupnými a detailnými informáciami. V prvých fázach projektu základné elementy softvérového procesu sú organizované v prvej úrovni štruktúry rozdelenia práce. Pracovné a plánované balíčky sú postupne začleňované do tejto štruktúry a postupnou realizáciou prác na projekte, ukončením prác v pracovných balíčkoch, sú plánované balíčky začlenené medzi pracovné a tým viac pracovných balíčkov je vložených do štruktúry rozdelenia práce. [5]

Analýza vytvorenej hodnoty založená na bodoch prípadov použitia sa dá použiť ako metrika pre projektové línie aj vývoj projektu. Tabuľka 3 porovnáva pojmy tradičnej analýzy vytvorenej hodnoty s analýzou vytvorenej hodnoty založenej na bodoch prípadov použitia.

Tab. 3. Porovnanie základných pojmov dvoch typov analýz vytvorenej hodnoty

Výraz	Tradičná analýza vytvorenej hodnoty	Analýza vytvorenej hodnoty založená na bodoch prípadov použitia
Základná línia pre výkon	Súčet odhadov plánu všetkých pracovných balíčkov (trvanie a vynaložené úsilie)	Celkový počet bodov prípadov použitia pre softvérový systém
Rozpočet pre projekt	Plánovaný rozpočet pre projekt	Prevedená peňažná hodnota plánovaných bodov prípadov použitia
Plánované percento dokončenia	Podiel peňažnej hodnoty kumulatívnych úloh a základnej línie pre výkon	Počet bodov prípadov použitia znásobený percentom pridelených tomu druhu úloh softvérového projektu
Skutočné percento dokončenia	Peňažná hodnota aktuálne dokončených pracovných balíčkov delená celkovou hodnotou rozpočtu pri ukončení	Podiel počtu ukončených bodov prípadov použitia a celkového plánovaného počtu bodov prípadov použitia

Štruktúra metódy monitorovania a kontroly softvérového projektu analýzou vytvorenej hodnoty je zobrazená na obrázku 2.



Obr. 2. Štruktúra metódy monitorovania analýzou vytvorenej hodnoty

Schéma začína analýzou charakteristík projektu. Rozdielne typy projektov vyžadujú rozličné procesné modely a prístupy k riadeniu. Následne je podľa charakteru projektu vybratý vhodný proces vývoja (prvá vetva obrázku 2) alebo splnené prípadné technické požiadavky. Navrhované používateľské požiadavky musia byť v prvom rade rozanalyzované a pretransformované do modelov prípadov použitia a tried analýzy. Po tejto fáze začína výpočet odhadov projektu, ktorý je potrebný, aby sa mohla vôbec začať sledovať práca na projekte a jeho vývoj, odhliadnuc od toho, aký presný bol počiatkový výpočet.

Vývoj softvéru nemožno rozdeliť na presné podúlohy s presným termínom ukončenia, pokiaľ nie je známy celkový dizajn systému. Preto sa odporúča, aby bol vývoj založený na dvoch druhoch plánov: fázový plán a iteratívny plán.

Fázový plán je hrubý plán, ktorý získava všeobecné informácie o projekte pre jeden cyklus. Zberá napríklad požadované zdroje a závislosti, pokiaľ sú nejaké známe. Iteračný

plán je podrobnejší a existuje jeden pre každú operáciu. Tento plán zahrňuje dôležité informácie, ako dátumy hlavných „buildov“ a revízií, ktoré slúžia potom štruktúre rozdelenia práce pri preformovaní plánovacích balíčkov na pracovné. Iteračný plán je zostavený pomocou bežných plánovacích techník a nástrojov na tvorbu úloh a ich pridelenie jednotlivým osobám pracujúcim na projekte.

Počas každého cyklu vývoja si každý člen tímu môže zvoliť vlastnú metódu rozdelenia sebe pridelených pracovných balíčkov na menšie podúlohy, aplikovať svoju vlastnú štruktúru rozdelenia práce tým, že svojím úlohám a podúlohám pridelí termíny začiatku a plánovaného ukončenia a takisto rozpočet pre dané úlohy. Pritom musí ale dbať na harmonogram vývoja celého projektu a svoju štruktúru rozdelenia práce tomu prispôbiť.

Projektové údaje stavov úloh, dátumy ukončenia a minulé financie sa pravidelne zberajú, vytvorené hodnoty sú konvertované do bodov prípadov použitia pre účel výpočtu parametrov analýzy vytvorenej hodnoty. Ak indexy odchýlky plánu alebo odchýlky nákladov naznačujú, že projekt odbočuje od stanovených cieľov, môžu byť použité metódy projektového manažmentu, napríklad zmena niektorých požiadaviek alebo preplánovanie úloh, aby sa projekt uberal opäť správnym smerom.

Po týchto krokoch je vypracovaný nový iteračný plán, vytvorené nové pracovné balíčky, rozvrh úloh, alokované nové zdroje a predefinovaná štruktúra rozdelenia práce. Projekt následne postupuje do novej iterácie.

Záver

V rámci eseje som stručne objasnil potrebu monitorovania softvérových projektov a v podstate projektov vo všeobecnosti. Následne som vysvetlil základné princípy jednotlivých metód a prístupov používaných pri monitorovaní softvérových projektov a to najmä analýzu vytvorenej hodnoty a metódu bodov prípadov použitia. Tieto dve metódy boli potom ďalšími predpokladmi na zvládnutie hlavnej časti eseje – spojenia analýzy vytvorenej hodnoty s metódou bodov prípadov použitia a je nasadenie na monitorovanie softvérových projektov. Evolučná štruktúra rozdelenia práce prispela k tomu, že projekty môžu byť monitorované a to takým spôsobom, že v procese vývoja sa postupne zohľadňujú zmeny, teda monitorovanie a plánovanie sa dá na základe tejto analýzy riadiť.

Cieľ eseje bol teda naplnený – aplikácia analýzy vytvorenej hodnoty sa dá použiť aj pri monitorovaní softvérových projektov. Možno už zostáva zodpovedať len jednu otázku? Kto z Vás, čo ste si prečítali túto esej, si trúfa monitorovať softvérový projekt analýzou vytvorenej hodnoty založenou na bodoch prípadov použitia?

Použitá literatúra

1. Anda, B., Dreiem, H., Sjoberg, D.I.K., Jorgensen, M.: Estimating Software Development Effort Based on Use Cases – Experiences from Industry. The 4th International Conference on the Unified Modeling Language, Concepts, and Tools (UML), Canada, October 1-5, 2001, 487-502.

2. Boehm, B.: Value-based software engineering: reinventing. ACM SIGSOFT Software Engineering Notes, Vol. 28, No. 2, ACM, New York (2003).
3. Fleming Q.W., J.M. Koppelman.: Earned Value Project Management. 3rd Edition, Project Management Institute, Atlanta, 2006.
4. Jones, Ch.: Making measurement work. CrossTalk, 2003. 9.10.2009 <http://www.psmc.com/Downloads/Other/MakingMeasurementWork.pdf>
5. Li, J., Ma, Z., Dong, H.: Monitoring software projects with earned value analysis and use case point. International Conference on Information Systems (2008), 475-480.
6. Schneider, G., Winters, J.P.: Applying Use Cases, Second Edition, Addison Wesley (2001).

Annotation

Monitoring software projects with earned value analysis

The earned value analysis (EVA) is a project management method for controlling project progress. There are two problems in applying this method on software projects. The first problem is how to consistently express project baselines and to measure technical progress of software projects. The second is due to the conventional Work Breakdown Structure that is prematurely structured around the product cannot reflect the software changes in the development. This essay devotes the problematics and solutions for using earned value analysis on software projects, presents an evolutionary work breakdown structure focusing on the software process elements, and describes the adoption of use case points as software project baselines and performance measurement. An approach to monitoring software project progress with earned value analysis and use case points is presented.