

Formularea obiectivelor si elaborarea structurii ierarhice a pachetelor de lucru (WBS)

I. Obiective

- Se va urmări modul în care pot fi formulate obiectivele SMART ale unui proiect și tehnicile de elaborare/verificare a WBS.

II. Breviar teoretic

Vezi note de curs (Capitolul 2 – accent pe subcapitole 2.1, 2.2.2, 2.2.3).

Aria de acoperire a unui proiect se numește „**scope**” (engl). Tot ceea ce un proiect își propune să facă aparține „scope”, ceea ce un proiect NU își propune să facă este exterior „scope”.

Primul lucru important în managementul de proiect este să înțelegi **CE AI DE FACUT**, altfel spus să înțelegi care este **rezultatul dorit**. Se spune că un proiect poate începe atunci când știi exact acest rezultat. De fapt, aceasta înseamnă să arăți care este „scope” –ul proiectului. Dacă acest rezultat este clar, înțeles, agreeat, poți ulterior să îți formezi o viziune despre cum poate fi obținut, să elaborezi alături de cei implicați un plan și să coordonezi executia planului, gestionând schimbările care intervin.

Dacă primul pas este gresit – descrierea rezultatelor dorite, atunci toată construcția (proiectul) poate deveni instabilă, nedorită, greu de gestionat.

Rezultatele sunt indicate prin formularea obiectivelor proiectului și, apoi, prin elaborarea unei structuri ierarhizate, denumită WBS.

Formularea obiectivelor – CE vrei să obții?

Rolul obiectivelor este de a sintetiza fără echivoc ce își propune proiectul să facă. Aceasta înseamnă că:

- formularile trebuie să fie cât mai scurte;
- formularile nu trebuie să fie interpretabile: trebuie evitați termenii care creează ambiguitate, formularile generale (sunt preferate particularizări, specificații tehnice, elemente concrete); este important aici de stabilit un echilibru între gradul de detaliere folosit și simplitatea în formulare;
- obiectivele trebuie să poată fi verificate dacă au fost îndeplinite; această verificare trebuie să fie simplă, dar mai ales, lipsită de subiectivism;
 - ⇒ Obiectivele se recomandă să fie criterii cuantificabile, definite prin atribut, metrică și valoare. Uneori însă acest lucru e dificil. Este suficient ca obiectivul să fie verificabil (ex: instalarea WinVista pe stația de lucru).
- obiectivele trebuie să fie realiste pentru compania care va derula proiectul sub aspectul logisticii, resurselor umane, resurselor financiare disponibile;
- obiectivele trebuie să poată fi atinse într-un interval rezonabil de timp – uzual sub 5 ani;
- obiectivele trebuie să poată fi realizate la nivelul actual de dezvoltare al societății/tehnicii/stiinței.

Aceste idei sunt sintetizate in acronimul “**SMART**”:

- „**S**pecific” = = precis, fara ambiguitati
- „**M**easurable” = = masurabil, verificabil - la sfârșitul proiectului/fazei să ne putem da seama dacă l-am atins sau nu
- „**A**chievable” = = sa poata fi atins (in contextul actual de cunoastere)
- „**R**ealistic” = = realist (pentru organizatie)
- „**T**imed” = = sa permita estimarea timpului necesar pentru indeplinirea sa

Obiectivele **trebuie agreate de toti cei implicati in proiect**, pentru ca e important ca rezultatele sa nu fie presupuneri ale managerului de proiect, ci chiar rezultatele dorite.

O tehnica simpla de formulare a obiectivelor este prezentata mai jos:

1. *Aflarea rezultatelor dorite.* Pentru determinarea rezultatelor trebuie sunt pornit de la a intelege ce schimbari se doresc fata de starea actuala (efectele dorite prin derularea proiectului). Astfel, se afla **CE** ar trebui facut pentru a provoca schimbarea solicitata, deci care sunt rezultatele dorite pe proiect.
2. *Gasirea indicatorilor de performanță adecvati si determinarea standardelor de performanță.*
Indicatorii (attribute, metrici) specifica ce performante sunt măsurate pentru a aprecia daca rezultatele dorite sunt obtinute (ex: timp maxim de raspuns [msec], eroare patratica de urmarire [unitati], numar teste esuate [unitati], numar utilizatori conectati simultan [unitati], etc).
Standardele de performanta sunt stabilite prin valorile tinta pentru indicatorii de performanță. Daca aceste standarde sunt indeplinite, rezultatul se considera atins.
Ex: 1000 utilizatori conectati simultan cu timp maxim de raspuns/cautare de 1sec.
3. *Determinarea timpului si costurilor necesare.*
La acest stadiu, aceste valori arata mai degraba in ce conditii proiectul ar avea sens (de ex, dpdv al profitabilitatii). Valorile sunt obtinute uzual printr-o estimare grosiera, de tip analogie/top-down. Aceste valori sunt rafinate ulterior, dupa ce se va raspunde si alte intrebari: CUM, CU CINE, CAND, etc., prin elaborarea planului proiectului.
4. *Formularea obiectivelor.* O formulare recomandata ar putea fi:
[Predicatl] (sa ...) [subiect] (atributul asociat indicatorului de performanță) al [grup țintă] până la [standard de performanță] într-un timp de [perioadă de timp] cu un cost total nu mai mare de [costul estimativ].
Ex: Sa creasca numarul de teste efectuate de echipa T1 la 100 teste/zi in 2 luni cu un cost de maxim 5.000 Euro.
Pentru indicatorul folosit in acest exemplu, atributul este nr de teste efectuate/zi, metrica este unitati, valoarea impusa prin standardul de performanta considerat este 100.
5. *Verificarea obiectivului:* Se verifica daca obiectivul ilustreaza complet si fara echivoc schimbarea pe care proiectul trebuie sa o produca. Obiectivele trebuie sa aiba aceeasi relevanta pentru toti cei implicati in proiect.

Din pacate, uzual e dificil de formulat obiective SMART inca din faza initiala a proiectului. Estimările de cost si timp (de la punctul 3) sunt grosiere. De asemenea, de multe ori gasirea unor indicatori de performanta care sa descrie complet rezultatul dorit este dificila. In domeniul IT

aceasta problema apare frecvent. De aceea, este de dorit ca macar obiectivul sa fie unul verificabil, daca nu masurabil.

O varianta buna de lucru este oferita de metoda „drill”: se formuleaza un obiectiv principal care este echivalentul unui „titlu” cat mai precis, complet de proiect si apoi se detaliaza acest obiectiv prin obiective secundare, organizate arborescent. Nu se recomanda includerea unui numar mare de nivele/ramuri, pentru ca toate aceste informatii pot fi suplimentar detaliate in WBS. Obiectivul principal nu este de obicei SMART, dar obiectivele-frunza indeplinesc, de regula, aceasta cerinta.

Pentru descrierea mai detaliata a „scope”-ului proiectului si, implicit a rezultatelor acestuia, pe langa formularea obiectivelor SMART se construiesc si o structura ierarhizata a pachetelor de lucru. Structura ierarhizata a pachetelor de lucru („*Work Breakdown Structure*” = WBS) indica **CE se va realiza in cadrul proiectului**: care sunt livrabilele majore („deliverables”), separarea pe subsisteme a produselor/serviciilor livrabile majore.

WBS rezulta prin **descompunere** si respecta toate cerintele impuse de aceasta tehnica.

Daca separarea se face dupa functionalitate, atunci WBS inseamna de fapt separarea fiecarui livrabil major in subsisteme componente (Fig. 1).

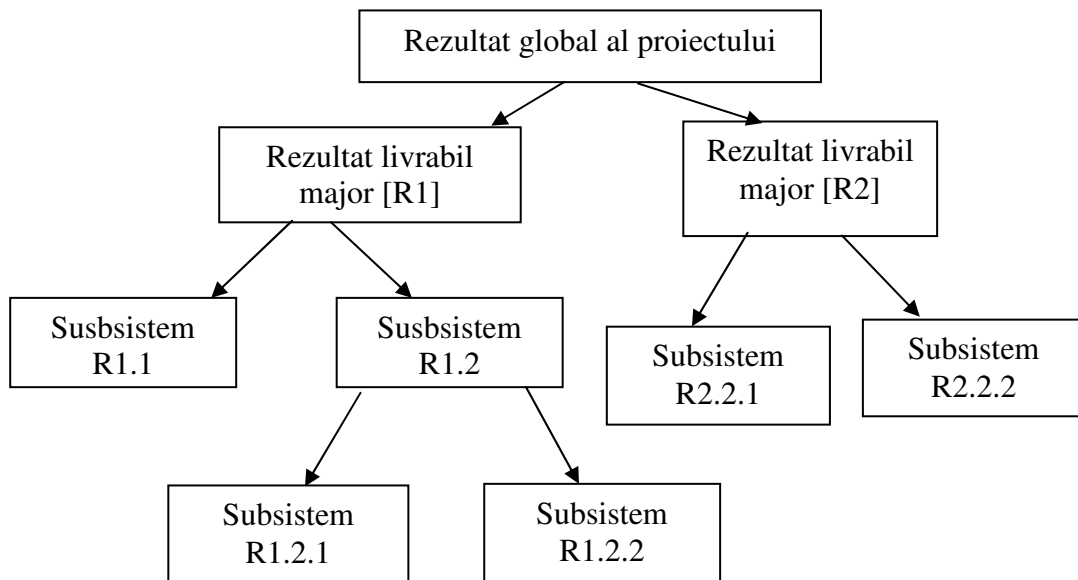


Fig. 1. WBS – o structura ierarhizata a rezultatelor proiectului

Frunzele din arborele WBS se numesc pachete de lucrări („work packages”). Important este ca acestea sa fie verificabile/masurabile si sa permita evidentiarea separata a resurselor necesare, duratei, bugetului. Fiecarui pachet de lucru i se poate asocia o activitate sau o lista de activitati corespunzatoare, avute in vedere la elaborarea detaliata a planului de lucru. Aceste activitati vor evidentia CUM poate fi obtinut respectivul rezultat.

Descompunerea ofera o imagine intuitiva asupra rezultatelor proiectului si permite intelegerea acestora pe gradul de detaliere dorit. Primele nivele sunt utile pentru managementul superior,

pentru o imagine de ansamblu, ultimele nivele elimina eventuale ambiguitati si ofera un instrument bun pentru a verifica daca rezultatele sunt indeplinite.

Metodele de verificare a WBS sunt detaliate in curs (subcapitole 2.1., 2.2.3).

III. Exemple

Pentru aplicatia descrisa in laboratorul 1 se ofera un exemplu de lucru. Trebuie inteles ca aceste exemple sunt doar sugestii, nu sunt tipare obligatoriu de folosit. Fiecare manager de proiect poate folosi un tipar pe care il considera mai elocvent pentru proiectul pe care il conduce.

Formularea obiectivelor:

O (Obiectiv principal): *Dezvoltarea unei aplicatii client-server care sa asigure inscrierea la cursuri si evaluarea studentilor dintr-o universitate, cu acces de la toate computerele aflate in campus.*

- O1 Aplicatia asigura portarea datelor existente in actuala baza de date
- O2 Aplicatia asigura urmatoarele functionalitati:
 - O2.1 selectarea ofertelor de curs de catre studenti, cu verificarea suprapunerii la orar, a restrictiilor impuse pe discipline, a numarului minim/maxim de studenti pe grupa; criteriul de departajare intre studenti fiind media obtinuta in semestrele anterioare
 - O2.2. evaluarea studentilor de catre profesori si vizualizarea notelor de catre studenti, profesori, secretar
 - O2.3. gestionarea informatiilor despre studenti, titluari, perioade de inscriere – evaluare de catre secretariat
 - O2.4. gestionarea userilor de catre administrator pentru acces securizat
- O3. Aplicatia va asigura:
 - O3.1 utilizare non-stop, cu perioade de intreruperi de maxim 10%
 - O3.2. acces simultan pentru maxim 2000 useri cu timp maxim de raspuns pe operatie de maxim 2 sec.
 - O3.3. compatibilitate Win NT, 2000, Vista.

Se observa ca aceste obiective au fost scrise pe baza cerintelor functionale/nonfunctionale preliminare.

Perioadele de timp si costurile nu au fost incluse, pentru simplitate, avand in vedere ca tehnicile de estimare timp/cost vor fi studiate in cursurile urmatoare.

Elaborarea WBS

WBS (pe model cascada, separare pe primele nivele dupa criteriul functionalitatii):

- 1. Aplicatia software
 - 1.1 Modul baza de date
 - 1.1.1 Structura tabelara
 - 1. 1. 1. 1 cerinte
 - 1. 1. 1. 2 design
 - 1. 1. 1. 3 implementare
 - 1. 1. 1. 4 testare
 - 1.1.2. Criptare date: cerinte, design, implementare, testare

- 1.1.3. Portare date vechi : cerinte, design, implementare, testare
- 1.2. Module de calcul + interfete
 - 1.2.1 Use-case login
 - 1. 2. 1. 1 cerinte
 - 1. 2. 1. 2 design
 - 1. 2. 1. 3 implementare
 - 1. 2. 1. 4 testare
 - 1.2. 2. Selectie cursuri studenti/profesori
 - 1.2. 3 Evaluare studenti de catre profesor
 - 1.2. 4 Menținere info profesori/studenti
 - se continua cu celelalte usecase-uri in mod similar
- 2. Documentatii
 - 2.1. User guide (detalii optiuni menu): cerinte, design, implementare, testare
 - 2. 2. Documentatie tehnica – ghid de instalare: cerinte, design, implementare, testare
 - 2. 3. Help on-line: cerinte, design, implementare, testare
 - 2.4. Tutorial – set de 5 lectii pentru prezentarea facilitatilor principale: cerinte, design, implementare, testare
- 3. Management de proiect

La aplicatiile software mai complexe se poate folosi descompunerea

- 1.1. Modul baza de date
- 1.2. Modul comunicare client-server
- 1.3. Modul interfete grafice utilizator-aplicatie (interfata log-in, selectie cursuri, etc)
- 1.4. Modul calcul

La sisteme embedded, primul nivel ar putea indica separarea pe modul software, modul hardware, modul mecanic etc

Acest tip de WBS este mai simplu de configurat si verificat, dar, dupa elaborare se recomanda o transformare a WBS intr-o forma care sa simplifice dezvoltarea planului, precum si monitorizarea si controlul executiei proiectului. Aceasta forma urmareste fazele de derulare a proiectului – la modelul de dezvoltare de tip cascada acestea sunt: cerinte, design, implementare, testare

WBS (model cascada, descompunere dupa fazele proiectului)

- I. Management proiect
- II. Aplicatie Software
 - II. 1. Cerinte
 - 1.1. Aplicatie software
 - 1.2. Documentatie
 - II.2. Design
 - 2. 1. Aplicatie software
 - design baze date
 - structura tabelara
 - procedura criptare parole
 - procedura portare date
 - modul calcul si interfete
 - definire clase-obiecte necesare

diagrame secventa pentru
 login
 mentinere info profi/studenti
 etc
 descrieri interfete grafice

2.2. Documentatie

II.3 Implementare

3.1 Aplicatie software

baze date
 structura tabelara
 procedura criptare parole
 procedura portare date
 modul calcul si interfete
 login
 mentinere info profi/studenti
 etc

3. 2. Documentatie

userguide
 guide instalare
 tutorial training
 help on-line

II.4 Testare

Testare de unitate pentru....
 Testare de integrare
 Testare de sistem-performante

IV. Mod de lucru

Pentru una din aplicatiile prezentate se vor parcurge urmatoorii pasi:

- elaborarea descrierii aplicatiei: cerinte functionale/nonfunctionale (vezi lab1);
- formularea obiectivelor principale si secundare prin metoda “drill”;
- realizarea WBS (separare dupa functionalitate si apoi separare dupa faze, pe model model prototip si model incremental).
- elaborarea preliminară a **planului de management al obiectivelor**

Indicatie:

Planul de management al obiectivelor arata care obiective sunt stabile, care obiective se pot schimba, cine va aproba schimbarea, eventual limite intre care se va accepta schimbarea.

Ca sugestie, se poate completa un tabel de tipul urmator:

<i>Obiectiv</i>	Grad stabilitate si senzitivitatea proiectului fata de aceasta schimbare	Cine e responsabil sa semnaleze schimbarea,	Cum se depisteaza schimbarea	Cine aproba schimbarea, care e procedura de schimbare	Variante de raspuns la schimbare