

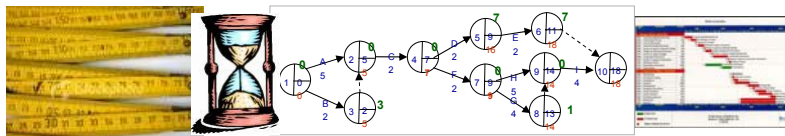


HY 351: Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων CS 351: Information Systems Analysis and Design

HY351:
Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων
Information Systems Analysis and Design



Διαχείριση Έργου (Project Management)



Γιάννης Τζιτζίκας

Διάλεξη : 5

Ημερομηνία : 27-10-2006

Θέμα :

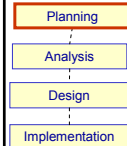


Διάρθρωση

- Τι είναι η Διαχείριση Έργου;
- Από ποια βήματα αποτελείται;
 - Εκτίμηση του Μεγέθους του Έργου
 - Δημιουργία και Διαχείριση του Πλάνου Εργασίας (WorkPlan)
 - Ανάλυση Κρίσιμου Μονοπατιού (Critical Path Analysis (CPA))
 - Διαγράμματα Gantt
 - Στελέχωση του Έργου (Staffing the Project)
 - Συντονισμός των Δραστηριοτήτων του Έργου (Coordinating activities)
 - Διαχείριση Επαναλήψεων (Managing iterations)
 - Χρονοπλαισίωση (Timeboxing)
- Άλλα Οργανωτικά Θέματα
- Συνηθισμένα Λάθη
- Εργαλεία για Διαχείριση Έργου



Επανάληψη



Στόχος της είναι η απάντηση των ερωτημάτων:

- Γιατί και πώς θα κατασκευαστεί το σύστημα;

Χωρίζεται σε δυο «υποφάσεις»:

(A) Έναρξη Έργου

- (την οποία συζητήσαμε στο 3ο μάθημα)
- Παραδοτέο: Μελέτη Σκοπιμότητας

(B) Αν το έργο εγκριθεί, τότε μπαίνει στη φάση της **Διαχείρισης Έργου** (Project Management) η οποία διαρκεί όσο και το έργο.

- Ο **συντονιστής του έργου** (project management) δημιουργεί και ενημερώνει το πλάνο εργασίας (work plan), στελεχώνει το έργο, και παρακολουθεί και ελέγχει την πρόοδο του.
Παραδοτέο: **Πλάνο Έργου** (project plan)



Κατανομή Πόρων και Προγραμματισμός (Resource Allocation and Planning)

Τα έργα ανάπτυξης πληροφορικής (όπως και κάθε άλλο έργο) έχουν ανάγκη ορθούς διαχείρισης για τη εξασφάλιση ότι θα ολοκληρωθούν εντός του προϋπολογισμένου κόστους και χρόνου.

- Τα μεγάλα έργα ανάπτυξης
 - εμπλέκουν πολλά διαφορετικά (και εξειδικευμένα) πρόσωπα.
 - αποτελούνται από δραστηριότητες των οποίων η σειρά είναι σημαντική.
- Εκ τούτου, υπάρχει ανάγκη για τεχνικές και εργαλεία στήριξης της διαδικασίας διαχείρισης έργου:
 - **Εκτίμηση** των χρημάτων, χρόνου, και προσωπικού που θα απαιτηθεί
 - Υποβοήθηση της διαδικασίας **αναθεώρησης** των εκτιμήσεων κατά τη διάρκεια εκτέλεσης του έργου
 - **Παρακολούθηση** και **διαχείριση** των εργασιών και δραστηριοτήτων που διενεργούνται από την ομάδα των κατασκευαστών.



Τι είναι η Διαχείριση Έργου ; (What is Project Management?)

- **Διαχείριση Έργου:** η διαδικασία προγραμματισμού και ελέγχου της ανάπτυξης ενός συστήματος με στόχο την κατασκευή ενός συστήματος με την ορθή λειτουργικότητα, μέσα στα προκαθορισμένα χρονικά και οικονομικά πλαίσια
- Πρόσωπο κλειδί: **Διαχειριστής Έργου (project manager)**



Από ποια βήματα αποτελείται η Διαχείριση Έργου;

- [A] Εκτίμηση του **Μεγέθους** του Έργου
- [B] Δημιουργία και Διαχείριση **Πλάνου Εργασίας (WorkPlan)**
- [Γ] **Στελέχωση** του Έργου (staffing)
- [Δ] **Συντονισμός** δραστηριοτήτων (**Coordinating activities**)



Από ποια βήματα αποτελείται η Διαχείριση Έργου;

- [A] Εκτίμηση του **Μεγέθους** του Έργου
- [B] Δημιουργία και Διαχείριση **Πλάνου Εργασίας (WorkPlan)**
- [Γ] **Στελέχωση** του Έργου (staffing)
- [Δ] **Συντονισμός** δραστηριοτήτων (**Coordinating activities**)



[A] Εκτίμηση του **Μεγέθους** του Έργου

Συνήθως βασίζεται στην **εμπειρία**

- (διότι δεν έχουμε ακόμα αναλύσει τις απαιτήσεις)
- => Θυμηθείτε το Επίπεδο 2 του CMM (4ο μάθημα)

Ένας **εμπειρικός κανόνας** για εκτίμηση του **χρόνου** που θα απαιτηθεί:

- Εκτίμηση του χρόνου περάτωσης του έργου, από το χρόνο που δαπανήθηκε στη φάση του προγραμματισμού (planning phase) του:

Συχνά η φάση του προγραμματισμού αντιστοιχεί στο 15% του συνολικού χρόνου.

- Συνάμα, μπορούμε να προσπαθήσουμε να **μετρήσουμε** διάφορες απόψεις του υπό κατασκευή συστήματος, χρησιμοποιώντας **μετρικές λογισμικού**, και βάσει των αποτελεσμάτων να κάνουμε κάποιες (ενδεχομένως ακριβέστερες) εκτιμήσεις.



Μετρικές Λογισμικού (*Software Metrics*)



Μετρική Λογισμικού:

- Μετρά μια ή περισσότερες απόψεις της παραγωγής λογισμικού

Τύποι μετρικών

Υπάρχουν πάρα πολλές. Ανάλογα με το αν γίνονται «εκ των προτέρων» ή «εκ των υστέρων», μπορούμε να τις διακρίνουμε σε:

- Μετρικές Αποτελέσματος (*result or control metrics*)
 - Χρησιμοποιούνται εκ των υστέρων
- Μετρικές Πρόβλεψης (*predictor metrics*)
 - Χρησιμοποιούνται εκ των προτέρων

Ανάλογα με το τι μετράνε μπορούμε να διακρίνουμε τις εξής γενικές κατηγορίες:

- Μετρικές Διαδικασίας (*Process metrics*)
 - Π.χ. πυκνότητα ελαττωμάτων, χρόνος διόρθωσης ελαττωμάτων, κλπ.
- Μετρικές Προϊόντος (*Product metrics*)
 - Π.χ. γραμμές κώδικα.



Ένα παράδειγμα χρήσης μετρικών για εκτίμηση του μεγέθους ενός έργου ακολουθεί



Μια προσέγγιση για την εκτίμηση του μεγέθους του έργου είναι αυτή των Πόντων Λειτουργίας (Function Points Approach)

Η γενική ιδέα:



Τα βήματα αυτής της προσέγγισης:

(α) Εκτίμηση μεγέθους του συστήματος

- (a1) εκτίμηση πόντων λειτουργίας (**function points**)
- (a2) εκτίμηση γραμμών κώδικα (lines of code)

(β) Εκτίμηση απαιτούμενης προσπάθειας (effort required)

- Σε ανθρωπομήνες (man months)

(γ) Εκτίμηση απαιτούμενου χρόνου

- Σε μήνες



(a1) εκτίμηση πλήθους πόντων λειτουργίας (Estimate Function Points)

Εκτίμηση του πλήθους των

- **inputs** (e.g. data entry screens)
- **outputs** (e.g. reports)
- **queries** (e.g. db queries)
- **files**
- **program interfaces**

Παραδείγματα

- **6 inputs**
- **19 outputs**
- **10 queries**
- **15 files**
- **3 program interfaces**

Παρατήρηση:

- Στην έναρξη του έργου (προφανώς) δεν γνωρίζουμε πόσα ακριβώς είναι. Κάνουμε εκτίμηση. Όσο εξελίσσεται το έργο αναθεωρούμε κατάλληλα. (δεν πρέπει να ξεχνάμε ότι η διαχείριση έργου είναι μια φάση που τελειώνει όταν τελειώνει το έργο).



(α1) εκτίμηση πολυπλοκότητας πόντων λειτουργίας (Estimate Function Points)

Κατόπιν εκτιμούμε την πολυπλοκότητα του κάθε πόντου
χαρακτηρίζοντας την ως:
μικρή / μέτρια / υψηλή

Παρατήρηση:

- Για την κατηγοριοποίηση αυτή δεν πρέπει να λάβουμε υπόψη τις ικανότητες της ομάδας εργασίας.

Παράδειγμα	Κατανομή βάσει πολυπλοκότητας
– 6 inputs	(3 low, 2 medium, 1 high)
– 19 outputs	(4 low, 10 medium, 5 high)
– 10 queries	(7 low, 0 medium, 3 high)
– 15 files	(0 low, 15 medium, 0 high)
– 3 program interfaces	(1 low, 0 medium, 2 high)



(α1.1) ποσοτικοποίηση της πολυπλοκότητας

– 19 outputs (4 low, 10 medium, 5 high)

**Πολλαπλασιάζουμε με ένα κατάλληλο παράγοντα
(π.χ. τα low με 5, τα medium με 5 και τα high με 7)**

Description	Low	Complexity Medium	High	Total
Inputs	— x 3	— x 4	— x 6	—
Outputs	4 x 4	10 x 5	5 x 7	101
Queries	— x 3	— x 4	— x 6	—
Files	— x 7	— x 10	— x 15	—
Program Interfaces	— x 5	— x 7	— x 10	—
TOTAL UNADJUSTED FUNCTION POINTS (TUFP)				338

Ανάλογα με τον τύπο της λειτουργίας μπορεί να έχουμε διαφορετικό συντελεστή (παρατηρείστε ότι οι συντελεστές για τις επερωτήσεις έχουν τους χαμηλότερους συντελεστές)

Άρα αυτή τη στιγμή η πολυπλοκότητα του συστήματος εκτιμάται ότι είναι 338 πόντοι



Κάθε ομάδα ενδέχεται να έχει μικρή ή μεγάλη εμπειρία/ ικανότητα σε ορισμένα θέματα.

Πως θα λάβουμε υπόψη τις ικανότητες της συγκεκριμένης ομάδας;



(α1.2) Ρυθμίζοντας την πολυπλοκότητα βάσει των ικανοτήτων της ομάδας έργου

Ένας τρόπος είναι να πολλαπλασιάσουμε του πόντους TUFP (που υπολογίσαμε προηγουμένως) με έναν συντελεστή (π.χ. από 0.5 έως 1.5) ανάλογα με τις ικανότητες της ομάδας.

Για παράδειγμα ένα απλός κανόνας για την επιλογή συντελεστών είναι ο εξής:

- (APC=) 0.65 για πολύ εύκολα για την ομάδα έργου συστήματα
- (APC=) 1 για συστήματα συνηθισμένης δυσκολίας (για την ομάδα έργου) συστήματα
- (APC=) 1.35 για δύσκολα/πολύπλοκα (για την ομάδα έργου) συστήματα

$$\mathbf{TAFP = TUFP * APC}$$

- TAFP: Total Adjusted Function Points
- TUFP: Total Unadjusted Function Points
- APC: Adjusted Processing Complexity

Παράδειγμα: Αν το έργο είναι εύκολο για την ομάδα τότε από τους 338 πόντους πάμε στους $338 * 0.65 = 220$ πόντους



Ένας πιο εκλεπτυσμένος τρόπος έκφρασης της εμπειρογνωμοσύνης της ομάδας έργου

Each team may have low or high expertise on some things.
We could capture this with a table of the following form:

	Scale of 1 to 3
Data Communications	0
Heavy Use Configuration	0
Transaction Rate	0
End-User efficiency	0
Complex Processing	0
Installation Ease	2
Multiple sites	0
Performance	0
Distributed functions	3
On-line data entry	0
On-line update	0
Reusability	0
Operational Ease	0
Extensibility	0
Project Complexity (PC)	5

0: no effect on processing complexity
...
3: great effect on processing complexity

This characterizes the expertise of a team

$APC = 0.65 + (0.01 * PC)$



(α2) Εκτίμηση γραμμών κώδικα:
Μετατρέποντας τους πόντους σε γραμμές κώδικα

Language	LOC/FunctionPoint
C	130
COBOL	110
JAVA	55
C++	50
Turbo Pascal	50
Visual Basic	30
PowerBuilder	15
HTML	15
Packages (e.g., Access, Excel)	10-40

Source: Capers Jones, Software Productivity Research

Παράδειγμα:

Άρα το έργο μας, που μέχρι τώρα έχει 220 πόντους, υλοποιηθεί σε C, τότε εκτιμάται ότι θα απαιτηθούν 28.600 γραμμές κώδικα. Αν υλοποιηθεί σε Java τότε 12.100 γραμμές.



(β). Εκτίμηση απαιτούμενης **προσπάθειας** (σε ανθρωπομήνες)

Εξαρτάται από το μέγεθος του συστήματος και τους ρυθμούς παραγωγής της ομάδας των προγραμματιστών.

Μια φόρμουλα για εκτίμηση των ανθρωπομηρών από το εκτιμώμενο πλήθος γραμμών κώδικα, είναι το μοντέλο COCOMO (W. Boehm). Για μικρού ή μέτριου μεγέθους έργα λογισμικού (π.χ. 100K γραμμές κώδικα από ομάδα δέκα ή λιγότερων προγραμματιστών), το μοντέλο αυτό προτείνει;

$$|\text{Ανθρωπομήνες}| = 1.4 * (|\text{Γραμμές κώδικα}|/1000)$$

Παράδειγμα:

Άρα το έργο μας, που σε Java εκτιμάται να έχει 12.100 γραμμές απαιτεί

$1.4 * 12.1 = 16.94$ ανθρωπομήνες.

Αυτό σημαίνει ότι ένας ανθρωπομήνας αντιστοιχεί σε περίπου 700 γραμμές κώδικα, άρα μια εργάσιμη ημέρα σε περίπου 35 γραμμές κώδικα.



(γ) Εκτίμηση Απαιτούμενου **Χρόνου** (σε μήνες) Με άλλα λόγια εκτίμηση της διάρκειας του έργου

Το ιστορικό αρχείο της ομάδας μπορεί να χρησιμοποιηθεί για τέτοιου είδους εκτιμήσεις. Ένας εμπειρικός κανόνας είναι ο εξής:

$$|\text{Μήνες}| = 3 * |\text{Ανθρωπομήνες}|^{(1/3)}$$

όπου **Μήνες** είναι ο αναμενόμενος χρόνος περάτωσης του έργου σε μήνες

- Παράδειγμα: για ένα έργο με εκτιμώμενη ανθρωποπροσπάθεια 14 μηνών, η αναμενόμενη διάρκεια του έργου είναι 7 μήνες
 - Σχόλιο: Παραλληλία και μέγιστη επιτάχυνση: Amdahl's Law



Από ποια βήματα αποτελείται η Διαχείριση Έργου;

- [Α] Εκτίμηση του **Μεγέθους** του Έργου
- [Β] Δημιουργία και Διαχείριση **Πλάνου Εργασίας (WorkPlan)**
- [Γ] **Στελέχωση** του Έργου (staffing)
- [Δ] **Συντονισμός** δραστηριοτήτων (Coordinating activities)



[Β] Δημιουργία και Διαχείριση **Πλάνου Εργασίας (WorkPlan)**

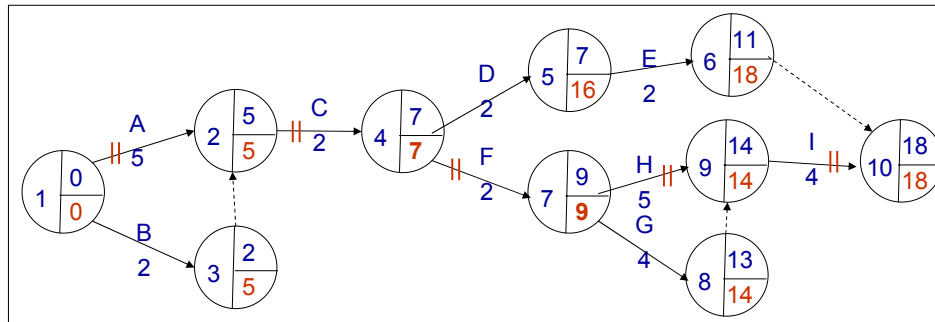
Εργαλεία και τεχνικές:

- **Ανάλυση Κρίσιμου Μονοπατιού**
 - **Critical Path Analysis (CPA)**
- **Διαγράμματα Gantt**
 - **Gantt charts**



Ανάλυση Κρίσιμου μονοπατιού Critical Path Analysis (CPA)

- Αρχικά αυτή η τεχνική ήταν γνωστή με το όνομα PERT (Project Evaluation and Review Technique)
 - Προτάθηκε στα τέλη της δεκαετίας του 50 για τα έργα ανάπτυξης οπλικών συστημάτων του Αμερικανικού Ναυτικού



Ανάλυση Κρίσιμου μονοπατιού

Βήματα

1. Απαρίθμηση όλων των δραστηριοτήτων και ορόσημων (milestones) του έργου.
2. Καθορισμός των εξαρτήσεων (dependencies) μεταξύ των δραστηριοτήτων
3. Εκτίμηση της διάρκειας της κάθε δραστηριότητας
4. Γραφική απεικόνιση του Διαγράμματος CPA



Ανάλυση Κρίσιμου μονοπατιού

1. Απαρίθμηση όλων των δραστηριοτήτων και ορόσημων (milestones) του έργου.

«*Milestone*»: Ορόσημο, πέτρα ένδειξης απόστασης, σημαντικό επίτευγμα

«*Ορόσημο*» (λεξικό Τεγόπουλου-Φυτράκη):

- Σημάδι που δείχνει το όριο έκτασης στην ξηρά ή στη θάλασσα
- Σύνορο
- Γεγονός που θεωρείται αφετηρία νέας περιόδου εξελίξεων ή προσπαθειών
 - Πχ: Το '21 αποτελεί ορόσημο για τον ελληνισμό, Ορόσημο στην ιστορία της τέχνης αποτελεί το έργο ...

Με ποιο τρόπο να ορίσουμε τις δραστηριότητες/ορόσημα του έργου;

Προσεγγίσεις

[1A] Από **Πάνω προς τα Κάτω** (Top-down)

[1B] Βάσει **Μεθοδολογίας** (δείτε 4^ο μάθημα)



Ανάλυση Κρίσιμου μονοπατιού

1. Απαρίθμηση όλων των δραστηριοτήτων και ορόσημων (milestones) του έργου.

[1A] Από **Πάνω προς τα Κάτω** (Top-down)

- Αρχικά εντοπίζουμε τις εργασίες σε υψηλό επίπεδο αφαίρεσης
- Κατόπιν τις αναλύουμε σε διαρκώς μικρότερες μονάδες
 - Το αποτέλεσμα αυτής της αποσύνθεσης αναφέρεται με το όνομα **work breakdown structure** (WBS) (.. «δομή αποσύνθεση εργασίας»)



Παράδειγμα WBS για το βάψιμο ενός δωματίου

(activity-oriented)

- Prepare materials
 - Buy paint
 - Buy a ladder
 - Buy brushes/rollers
 - Buy wallpaper remover
- Prepare room
 - Remove old wallpaper
 - Remove detachable decorations
 - Cover floor with old newspapers
 - Cover electrical outlets/switches with tape
 - Cover furniture with sheets
- Paint the room
- Clean up the room
 - Dispose or store left over paint
 - Clean brushes/rollers
 - Dispose of old newspapers
 - Remove covers

(deliverable-oriented)

- Material Preparation
 - Paint preparation
 - Ladder preparation
 - Brushes/rollers preparation
 - Wallpaper Remover
- Room Preparation
 - Old wallpaper removal
 - Detachable decorations removal
 - Floor protection
 - Electrical outlets/switches protection
 - Furniture protection
- Room Painting
- Room cleanup
 - Leftover paint disposal
 - Brushes/rollers cleaning
 - Old newspapers disposal
 - Covers removal



Παράδειγμα WBS από ένα ευρωπαϊκό έργο πληροφορικής

SP 1 Methodology, standards and existing resources		
WP	1100	Review state of the art
	###	Methodology review
	###	Research and technological existing solution
	###	Existing Standard for virtualisation / preservation
Tasks		
WP	1200	CASPAR methodology and models
	###	Caspar overall methodology
	###	Caspar guidelines
	###	Contribution to standards (ex. OASIS extension)
	###	Conceptual model
	###	Simple digital objects virtualisation modelling
	###	Complex objects virtualisation modelling
	###	On-demand objects modelling
Tasks	###	Certification
WP	1300	Overall architecture
	###	Overall Component Architecture design
	###	Component model
SP 2 Component level Research & Technology Development		
WP	2100	Storage
	###	Storage technology research

- More on CASPAR folder:

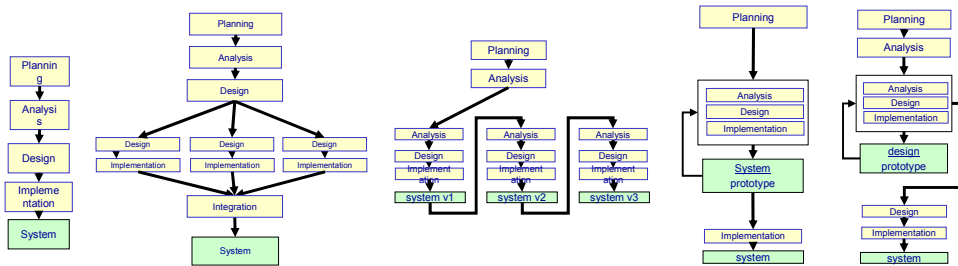


Ανάλυση Κρίσιμου μονοπατιού

1. Απαρίθμηση όλων των δραστηριοτήτων και ορόσημων (milestones) του έργου.

[1B] Βάσει Μεθοδολογίας (δείτε 4^ο μάθημα)

- Χρησιμοποιούμε μια λίστα εργασιών και ορόσημων από μια γενική μεθοδολογία (π.χ. Καταρράκτη, Παράλληλη, Πολυφασική, RUP, ...)



Παράδειγμα

1. Απαρίθμηση όλων των δραστηριοτήτων και ορόσημων (milestones) του έργου.

Activity	Description	Milestone
A	Interview users	2
B	Prepare use cases	3
C	Review use cases	4
D	Draft screen layouts	5
E	Review screens	6
F	Identify classes	7
G	CRC Analysis	8
H	Prepare draft class diagram	9
I	Review class diagram	10



Παράδειγμα καρτέλας δραστηριότητας

Με περισσότερα στοιχεία:

Work Plan Information	Example
Name of activity	B. Prepare Use Cases
Start date	Oct 11, 2005
Completion date	Oct 20, 2005
Person assigned	Student X
Deliverable(s)	Use Cases and Use Case Diagrams
Completion status	Open
Priority	High
Resources needed	Spreadsheet
Estimated time	4 hours
Actual time	2 hours



Παράδειγμα

1. Απαρίθμηση όλων των δραστηριοτήτων και ορόσημων (milestones) του έργου.
2. Καθορισμός των εξαρτήσεων (dependencies) μεταξύ των δραστηριοτήτων

Activity	Description	Milestone	Preceding activities	Expected duration	Staffing
A	Interview users	2	-	5	2
B	Prepare use cases	3	-	2	See A
C	Review use cases	4	A, B	2	3
D	Draft screen layouts	5	C	2	2
E	Review screens	6	D	2	2
F	Identify classes	7	C	2	3
G	CRC Analysis	8	F	4	3
H	Prepare draft class diagram	9	F	5	3
I	Review class diagram	10	G, H	4	4



Παράδειγμα

- Απαρίθμηση όλων των δραστηριοτήτων και ορόσημων (milestones) του έργου.
 2. Καθορισμός των εξαρτήσεων (dependencies) μεταξύ των δραστηριοτήτων
 3. Εκτίμηση της διάρκειας της κάθε δραστηριότητας

Ένας εμπειρικός κανόνας που μπορούμε να χρησιμοποιήσουμε:

Αναμενόμενη διάρκεια μιας εργασίας:

$$ED = (MOT + MPT + 4*MLT)/6$$

- MOT: most optimistic time for the completion of an activity
 – Υπόθεση: δεν θα υπάρξει καμία καθυστέρηση
- MLT: most likely time for the completion of an activity
- MPT: most pessimistic time for the completion of an activity
 – Υπόθεση: «όλα» θα πάνε στραβά



Παράδειγμα

- Απαρίθμηση όλων των δραστηριοτήτων και ορόσημων (milestones) του έργου.
 2. Καθορισμός των εξαρτήσεων (dependencies) μεταξύ των δραστηριοτήτων
 3. Εκτίμηση της διάρκειας της κάθε δραστηριότητας

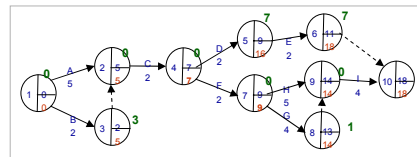
Activity	Description	Milestone	Preceding activities	Expected duration
A	Interview users	2	-	5
B	Prepare use cases	3	-	2
C	Review use cases	4	A, B	2
D	Draft screen layouts	5	C	2
E	Review screens	6	D	2
F	Identify classes	7	C	2
G	CRC Analysis	8	F	4
H	Prepare draft class diagram	9	F	5
I	Review class diagram	10	G, H	4



Διάγραμμα CPA

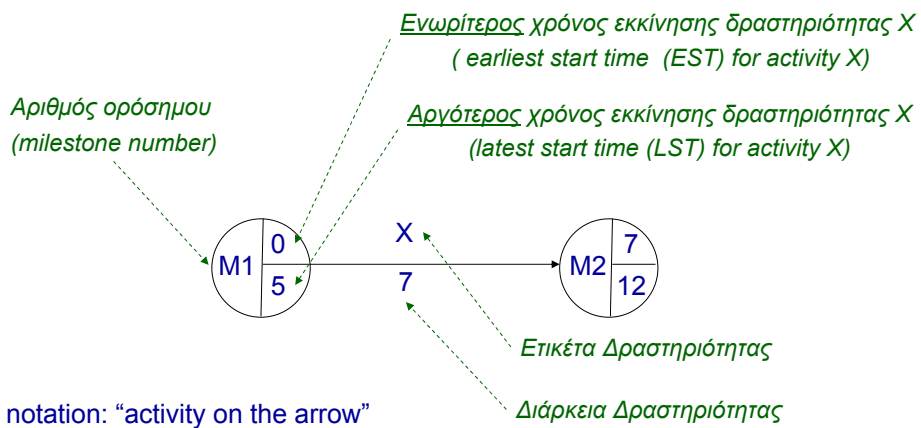
1. Απαρίθμηση όλων των δραστηριοτήτων και ορόσημων (milestones) του έργου.
2. Καθορισμός των εξαρτήσεων (dependencies) μεταξύ των δραστηριοτήτων
3. Εκτίμηση της διάρκειας της κάθε δραστηριότητας
4. Γραφική απεικόνιση του Διαγράμματος CPA

Activity	Description	Milestone	Preceding activities	Expected duration	Staff
A	Interview users	2	-	5	2
B	Prepare use cases	3	-	2	See A
C	Review use cases	4	A, B	2	3
D	Draft screen layouts	5	C	2	2
E	Review screens	6	D	2	2
F	Identify classes	7	C	2	3
G	CRC Analysis	8	F	4	3
H	Prepare draft class diagram	9	F	5	3
I	Review class diagram	10	G, H	4	4



Διάγραμμα CPA

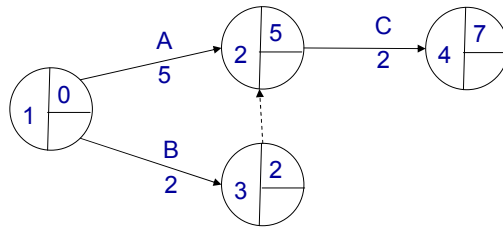
1. Απαρίθμηση όλων των δραστηριοτήτων και ορόσημων (milestones) του έργου.
2. Καθορισμός των εξαρτήσεων (dependencies) μεταξύ των δραστηριοτήτων
3. Εκτίμηση της διάρκειας της κάθε δραστηριότητας
4. Γραφική απεικόνιση του Διαγράμματος CPA





Παράδειγμα :
Συμπληρώνοντας τους ενωρίτερους χρόνους στο διάγραμμα CPA

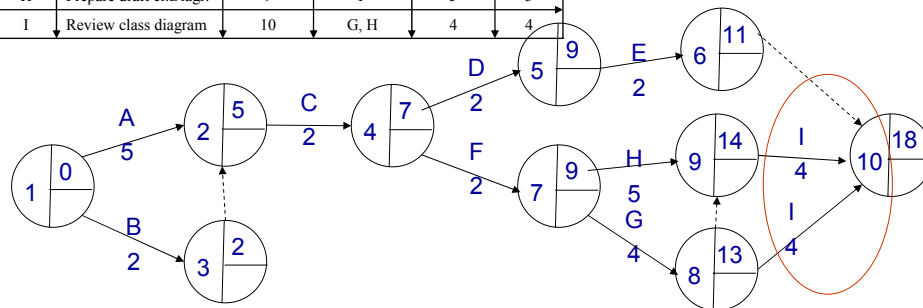
Acti- vity	Description	Milestone	Preceding activities	Expected duration	Staffin g
A	Interview users	2	-	5	2
B	Prepare use cases	3	-	2	See A
C	Review use cases	4	A, B	2	3
D	Draft screen layouts	5	C	2	2
E	Review screens	6	D	2	2
F	Identify classes	7	C	2	3
G	CRC Analysis	8	F	4	3
H	Prepare draft cl.Diagr.	9	F	5	3
I	Review class diagram	10	G, H	4	4



Παράδειγμα :
Συμπληρώνοντας τους ενωρίτερους χρόνους στο διάγραμμα CPA

Acti- vity	Description	Milestone	Preceding activities	Expected duration	Staffin g
A	Interview users	2	-	5	2
B	Prepare use cases	3	-	2	See A
C	Review use cases	4	A, B	2	3
D	Draft screen layouts	5	C	2	2
E	Review screens	6	D	2	2
F	Identify classes	7	C	2	3
G	CRC Analysis	8	F	4	3
H	Prepare draft cl.Diagr.	9	F	5	3
I	Review class diagram	10	G, H	4	4

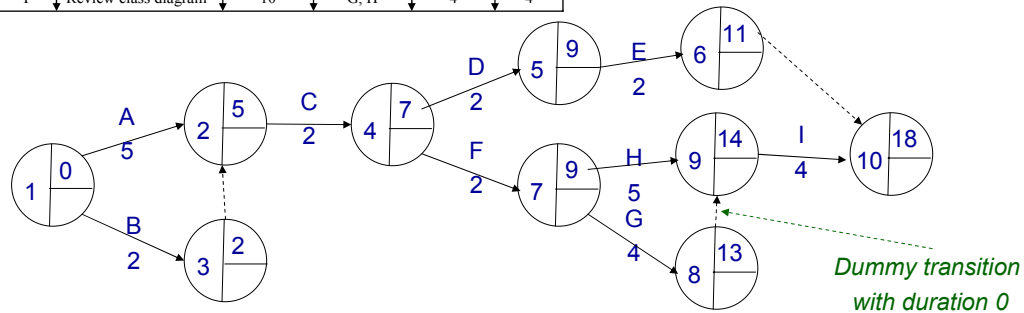
forward pass





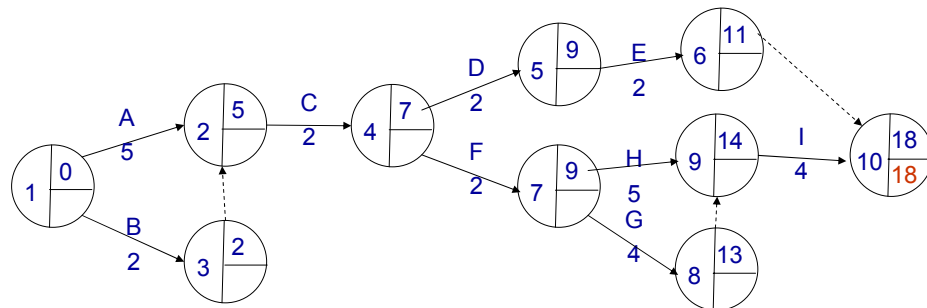
Παράδειγμα :
 Συμπληρώνοντας τους ενωρίτερους χρόνους στο διάγραμμα CPA

Activity	Description	Milestone	Preceding activities	Expected duration	Staffing
A	Interview users	2	-	5	2
B	Prepare use cases	3	-	2	See A
C	Review use cases	4	A, B	2	3
D	Draft screen layouts	5	C	2	2
E	Review screens	6	D	2	2
F	Identify classes	7	C	2	3
G	CRC Analysis	8	F	4	3
H	Prepare draft cl.Diagr.	9	F	5	3
I	Review class diagram	10	G, H	4	4



Παράδειγμα :
 Συμπληρώνοντας τους αργότερους χρόνους στο διάγραμμα CPA

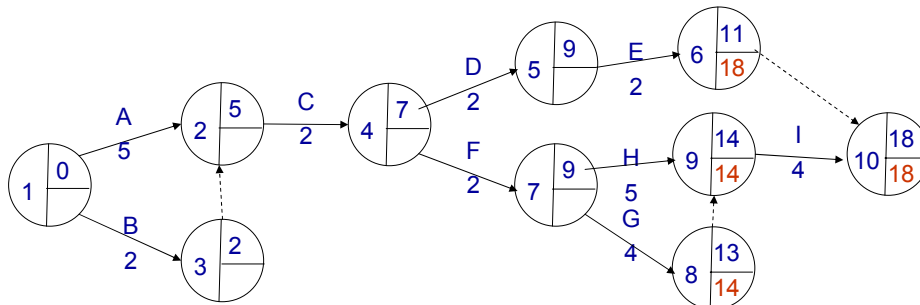
backwards pass





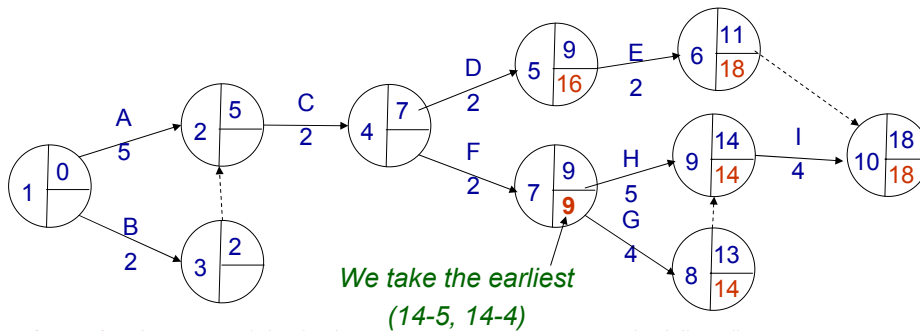
Παράδειγμα :
 Συμπληρώνοντας τους αργότερους χρόνους στο διάγραμμα CPA

backwards pass



Παράδειγμα :
 Συμπληρώνοντας τους αργότερους χρόνους στο διάγραμμα CPA

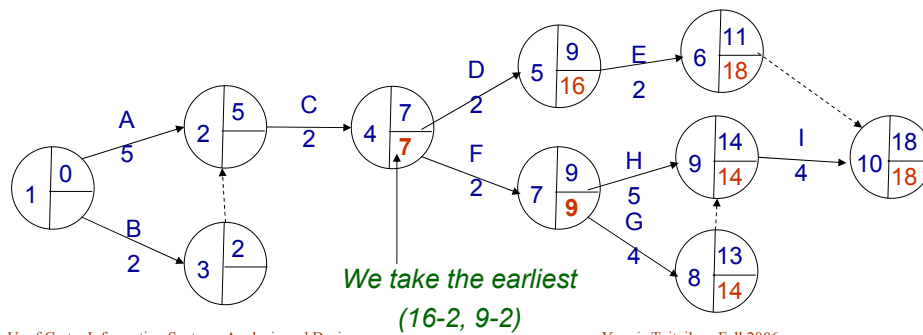
backwards pass





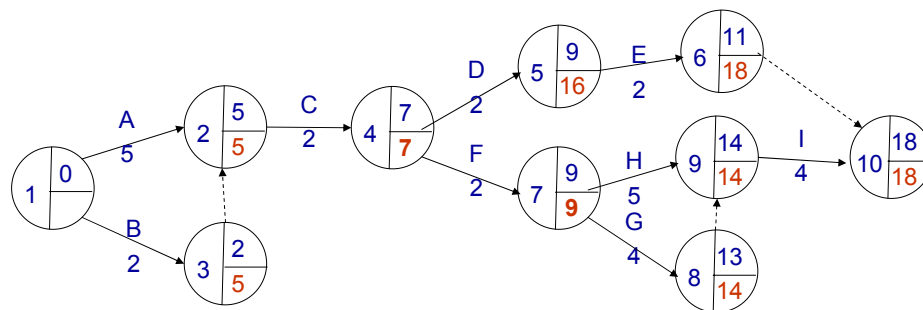
Παράδειγμα :
 Συμπληρώνοντας τους αργότερους χρόνους στο διάγραμμα CPA

backwards pass



Παράδειγμα :
 Συμπληρώνοντας τους αργότερους χρόνους στο διάγραμμα CPA

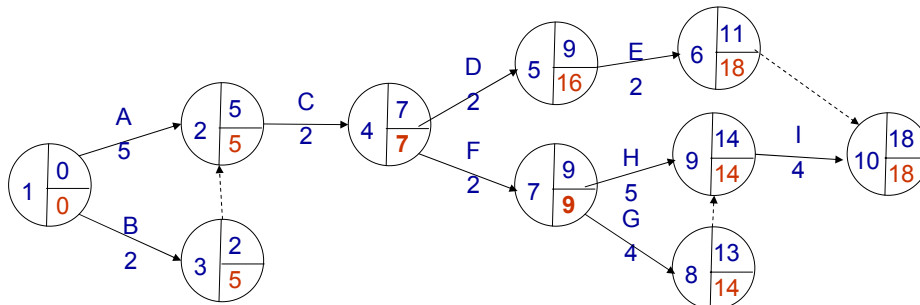
backwards pass





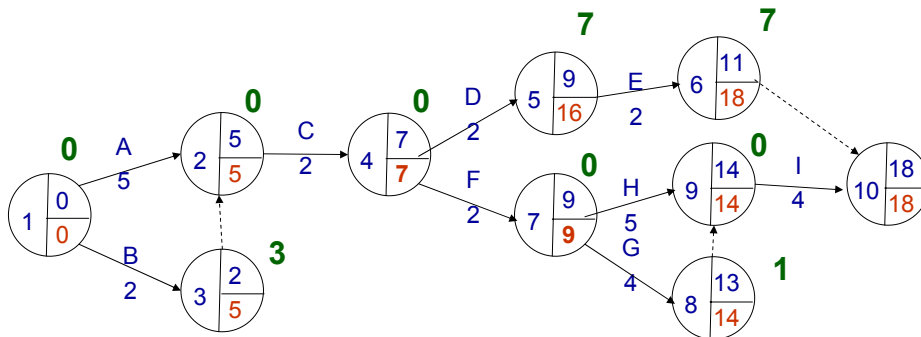
Παράδειγμα :
Συμπληρώνοντας τους αργότερους χρόνους στο διάγραμμα CPA

backwards pass



Αδρανής χρόνος μιας δραστηριότητας
(Slack (or float) time of an activity)

- Η διαφορά μεταξύ ενωρίτερου και αργότερου χρόνου εκκίνησης
- Εκφράζει το μέγιστο χρόνο καθυστέρησης που μπορεί να έχει μια δραστηριότητα χωρίς να επηρεάσει (επιμηκύνει) την διάρκεια ολόκληρου του έργου

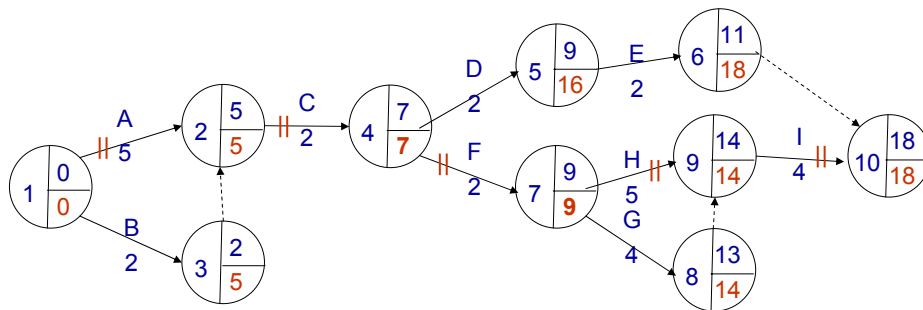




Εντοπίζοντας το κρίσιμο μονοπάτι (identifying the critical path)

Κρίσιμο Μονοπάτι:

Το μονοπάτι μέσω ορόσημων που έχουν αδρανή χρόνο 0.
(the path through all milestones that have a slack time of 0).



Συνοψίζοντας το CPA

• Διαγράμματα CPA: **Όφελος**

- Τα διαγράμματα CPA αποτελούν ένα αποτελεσματικό εργαλείο για να εντοπίσουμε τις δραστηριότητες των οποίων η ολοκλήρωση είναι **κρίσιμη** για την έγκαιρη ολοκλήρωση του έργου.
- Οποιαδήποτε καθυστέρηση δραστηριότητας εκ της οποίας διέρχεται το κρίσιμο μονοπάτι έχει ως συνέπεια την καθυστέρηση ολόκληρου του έργου

• Διαγράμματα CPA: **Αδυναμία**

- Δεν εικονογραφούν τις επικαλύψεις μεταξύ των δραστηριοτήτων (τα διαγράμματα Gantt είναι καλύτερα σε αυτό το θέμα)



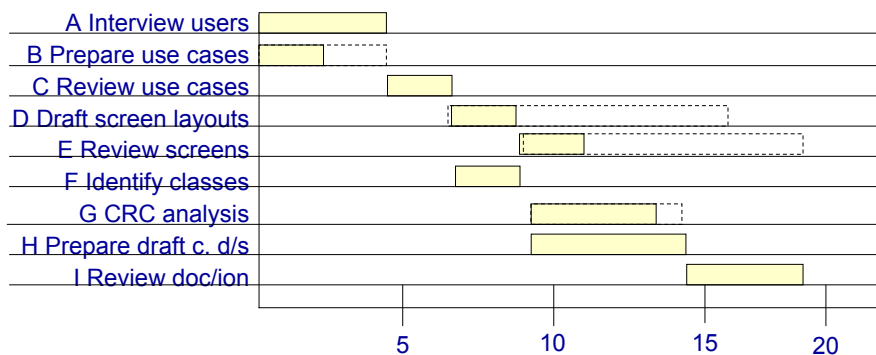
Διαγράμματα Gantt (Gantt charts)

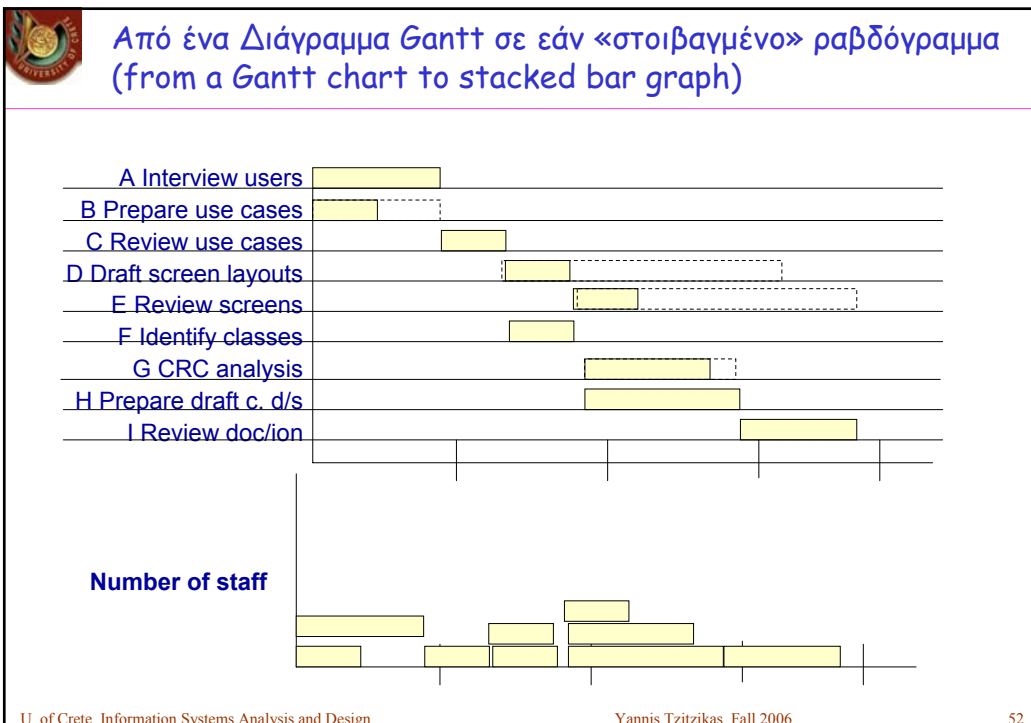
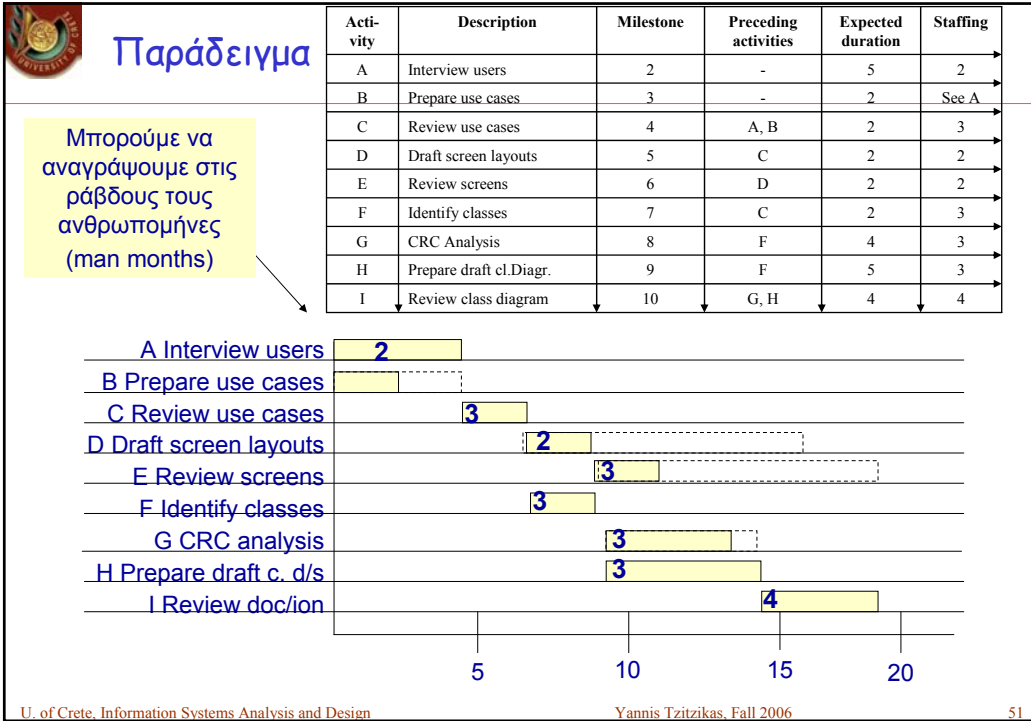
- Οριζόντιος άξονας: Χρόνος
- Κάθετος άξονας: δραστηριότητες
- Για κάθε δραστηριότητα ζωγραφίζουμε μια ράβδο (bar)
 - Μήκος ράβδου: αναμενόμενη διάρκεια της δραστηριότητας
 - Με διάστικτες (dashed) γραμμές/ράβδους μπορούμε να απεικονίσουμε τον αδρανή χρόνο (slack time) της κάθε δραστηριότητας.



Παράδειγμα

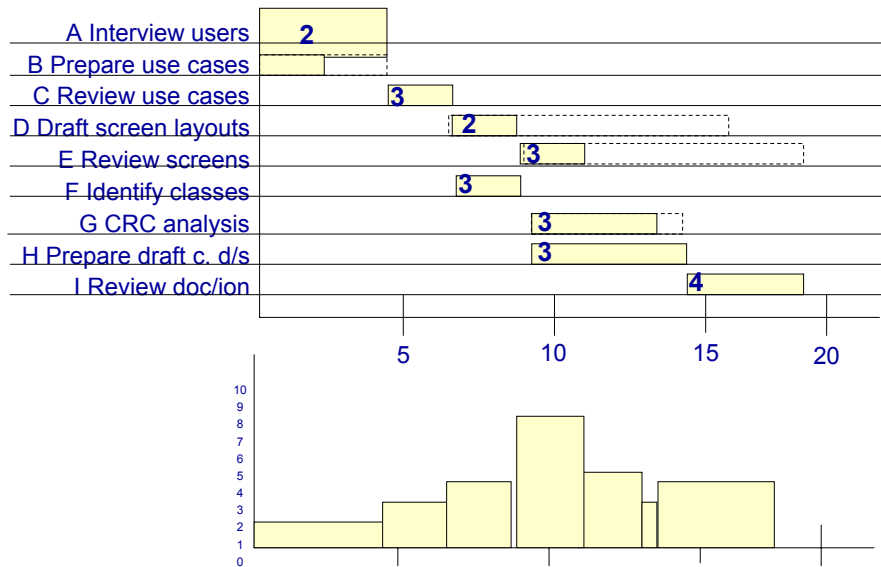
Acti- vity	Description	Milestone	Preceding activities	Expected duration	Staffing
A	Interview users	2	-	5	2
B	Prepare use cases	3	-	2	See A
C	Review use cases	4	A, B	2	3
D	Draft screen layouts	5	C	2	2
E	Review screens	6	D	2	2
F	Identify classes	7	C	2	3
G	CRC Analysis	8	F	4	3
H	Prepare draft cl.Diagr.	9	F	5	3
I	Review class diagram	10	G, H	4	4







Από ένα Διάγραμμα Gantt σε είν «στοιβαγμένο» ραβδόγραμμα



(μερικά παραδείγματα Gantt διαγραμμάτων
από πραγματικά έργα)

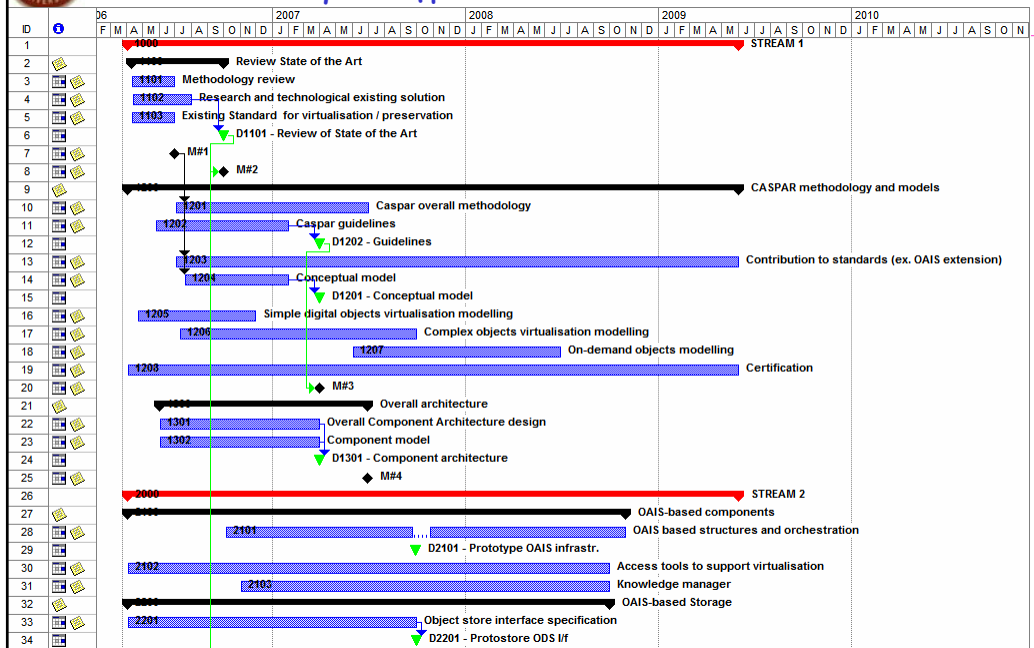


Παράδειγμα

Α/α	Περιγραφή δραστηριότητας	1	2	3	4	5	6
ΠΕ1	Ανάπτυξη συστήματος ηλεκτρονικής τεκμηρίωσης και θεματικής ταξινόμησης ιστορικών εγγράφων						
	ΠΕ1.1: Μελέτη εφαρμογής, προσδιορισμός απαιτήσεων						
	ΠΕ1.2: Σχεδιασμός εξειδικευμένων μοντέλων		Π1				
	ΠΕ1.3: Παραμετροποίηση εργαλείου διαχείρισης ορολογίας		Π2				
	ΠΕ1.4: Παραμετροποίηση εργ. ηλεκτρονικής, τεκμηρίωσης					Π3	
	ΠΕ1.5: Παραμετροποίηση εργαλείου θεματικής ταξινόμησης					Π4	
ΠΕ2	Ανάπτυξη εφαρμογής για τη δημιουργία δικτυακού τόπου πληροφόρησης						
	ΠΕ2.1: Μελέτη εφαρμογής, προσδιορισμός απαιτήσεων						
	ΠΕ2.2: Σχεδιασμός εξειδικευμένων μοντέλων						
	ΠΕ2.3: Δημιουργία δικτυακού τόπου πληροφόρησης					Π5	
U. of Crete	ΠΕ2.4: Δημιουργία συστήματος Αρχιτεκτονικών Διαμορφώσεων					Π6	



Ένα άλλο παράδειγμα





Από ποια βήματα αποτελείται η Διαχείριση Έργου;

- [Α] Εκτίμηση του **Μεγέθους** του Έργου
- [Β] Δημιουργία και Διαχείριση **Πλάνου Εργασίας (WorkPlan)**
- [Γ] **Στελέχωση** του Έργου (staffing)
- [Δ] **Συντονισμός** δραστηριοτήτων (Coordinating activities)



[Γ] **Στελέχωση** του Έργου (staffing)

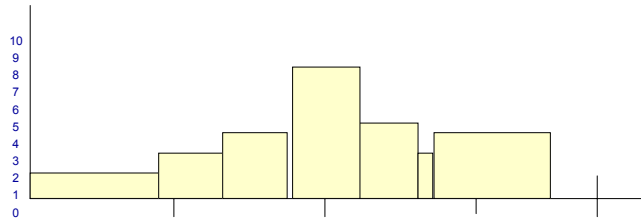
Σχετικά ερωτήματα:

- **Πόσα άτομα πρέπει να εργαστούν για αυτό το έργο;**
- **Πώς να κατανείμουμε ρόλους και αρμοδιότητες στα μέλη της ομάδας;**



Στελέχωση Έργου και Διαγράμματα Gantt (Staffing the Project and Gantt charts)

Τα άτομα που θα χρειαστούμε σε κάθε χρονική στιγμή του έργου φαίνονται και από το διάγραμμα Gantt.



Πολλές φορές αξίζει να προσπαθήσουμε να κάνουμε «Εξομάλυνση» πόρων (Resource smoothing): «Παίζουμε» με τους αδρανείς χρόνους για να ομαλοποιήσουμε (και να μειώσουμε) το πλήθος των ατόμων που πρέπει να προσληφθούν.



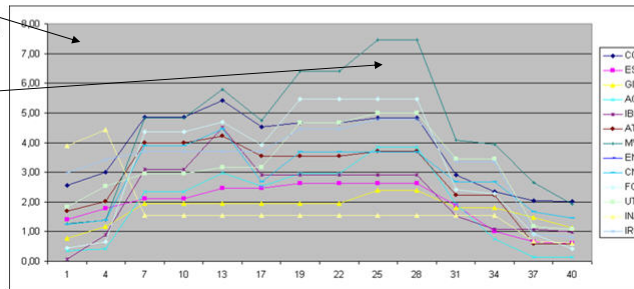
Παράδειγμα: Στελέχωση έργων με πολλούς εταίρους

Εκτίμηση ανθρώπινων πόρων που απαιτείται από κάθε εταίρο.

Στοιβαγμένο ραβδόγραμμα Gantt για κάθε εταίρο ξεχωριστά.

Τρίμηνο	1	4	7	10	13	17	19	22	25	28	31	34	37	
Partner1	2,55	3,01	4,86	4,86	5,41	4,52	4,67	4,67	4,83	4,83	2,90	2,35	2,04	2
Partner2	1,41	1,79	2,12	2,12	2,45	2,45	2,62	2,62	2,62	2,62	1,87	1,02	0,66	0
Partner3	0,77	1,18	1,95	1,95	1,95	1,95	1,95	2,40	2,40	1,81	1,81	1,48	1	0
Partner4	0,35	0,42	2,36	2,36	2,97	2,52	2,95	2,95	3,84	3,84	1,92	0,74	0,14	0
Partner5	0,08	0,88	3,11	3,11	4,50	2,91	2,91	2,91	2,91	2,91	1,52	1,08	1,08	0
Partner6	1,68	2,02	3,99	3,99	4,21	3,54	3,54	3,71	3,71	3,71	2,24	2,24	0,59	0
Partner7	1,27	1,38	4,84	4,84	5,79	4,73	6,40	6,40	7,45	7,45	4,09	3,84	2,65	1
Partner8	1,25	1,38	3,90	3,90	4,46	2,68	3,68	3,68	3,68	3,68	2,66	2,66	1,67	1
Partner9	1,25	1,38	3,90	3,90	4,46	2,68	3,68	3,68	3,68	3,68	2,66	2,66	1,67	1
Partner10	0,45	0,66	4,35	4,35	4,69	3,91	5,46	5,46	5,46	5,46	2,42	2,27	0,92	0
Partner11	1,84	2,53	2,95	2,95	3,17	3,17	4,67	4,67	5,00	5,00	3,45	3,45	1,17	1
Partner12	3,90	4,44	1,55	1,55	1,55	1,55	1,55	1,55	1,55	1,55	1,55	1,55	0,69	0
Partner13	2,98	3,42	3,70	3,70	3,70	3,70	4,45	4,45	4,78	4,78	3,36	3,36	0,92	0

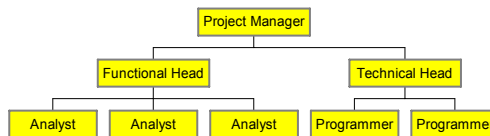
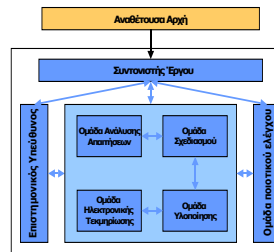
Αυτό είναι ένα σημείο στο οποίο αξίζει να προσπαθήσουμε να κάνουμε εξομάλυνση



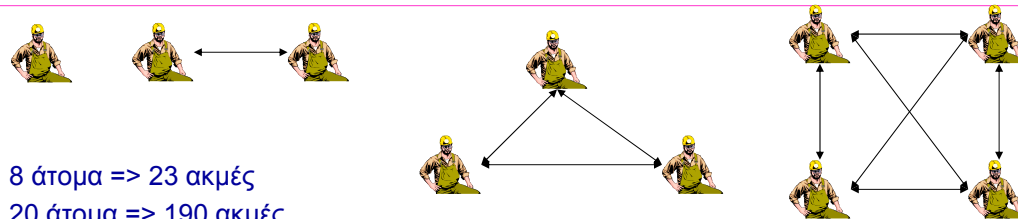


Πώς να κατανεύουμε ρόλους και αρμοδιότητες στα μέλη της ομάδας;

- (στο 2^ο μάθημα μιλήσαμε για ρόλους της ομάδας έργου)
- Πρέπει να ταιριάζουμε τις ικανότητες των προσώπων (που έχουμε ή πρόκειται να προσλάβουμε) με τις ανάγκες του έργου
- Πρέπει να ορίσουμε τον τρόπο με τον οποίο η ομάδα επικοινωνεί και θα λαμβάνει αποφάσεις
 - Reporting structure



Όσο μεγαλώνει η ομάδα τόσο πιο πολύπλοκη και ακριβή γίνεται η επικοινωνία και η λήψη αποφάσεων

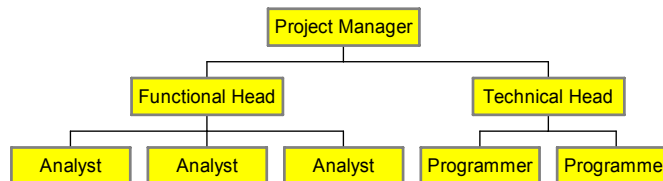


8 άτομα => 23 ακμές

20 άτομα => 190 ακμές

$$|Persons| * (|Persons| - 1) / 2$$

Για αυτό είναι απαραίτητη μια οργανωτική δομή





Άλλα ζητήματα σχετικά με την ομαδική εργασία

- Εύρεση τρόπων ενθάρρυνσης/υποκίνησης των μελών ομάδας να επιτύχουν τους στόχους του έργου
- Ελαχιστοποίηση συγκρούσεων/διαμαχών μεταξύ των μελών της ομάδας
 - Ένας τρόπος να μειώσουμε τις πιθανές συγκρούσεις είναι να εκχωρήσουμε συγκεκριμένους ρόλο σε κάθε μέλος της ομάδας.

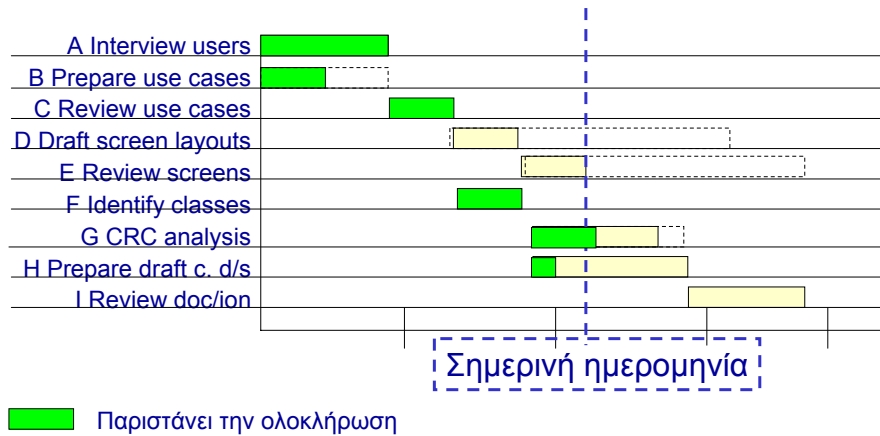


Από ποια βήματα αποτελείται η Διαχείριση Έργου;

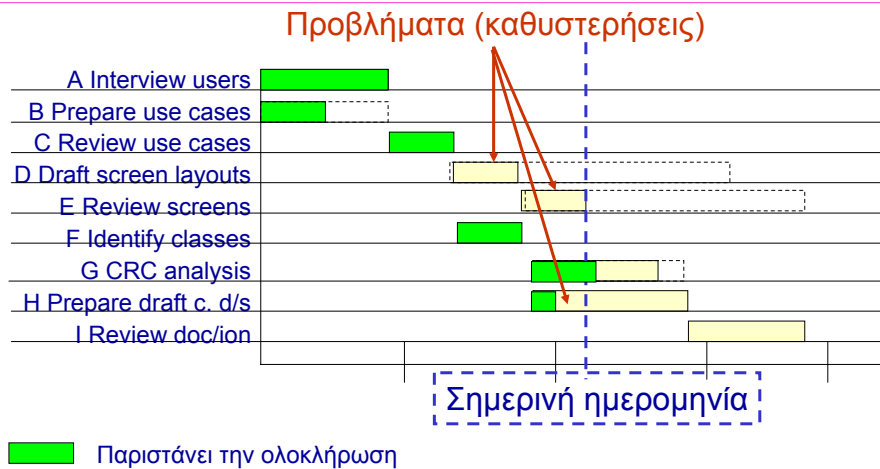
- [A] Εκτίμηση του Μεγέθους του Έργου
- [B] Δημιουργία και Διαχείριση Πλάνου Εργασίας(WorkPlan)
- [Γ] Στελέχωση του Έργου (staffing)
- [Δ] Συντονισμός δραστηριοτήτων (Coordinating activities)



Παρακολούθηση προόδου του έργου

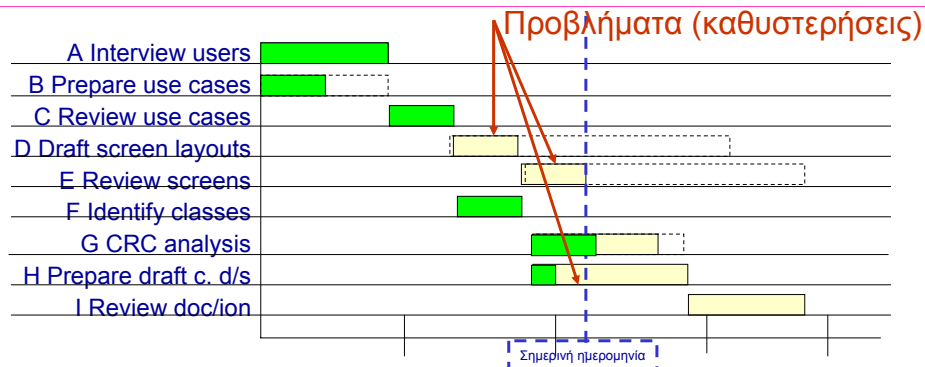


Εντοπισμός καθυστερήσεων





Έλεγχος του Έργου (αντιμετώπιση καθυστερήσεων)



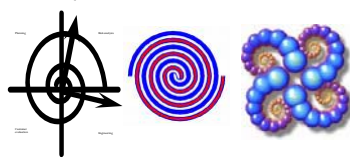
Πιθανοί τρόποι αντιμετώπισης των προβλημάτων

- Εκμετάλλευση των αδρανών χρόνων, αν αυτό είναι εφικτό (στο D,E είναι, στο H όχι)
- Περισσότερο προσωπικό (δεν είναι πάντα καλό, π.χ. επισκευή αυτοκινήτου)
- Καθυστέρηση του έργου
- Μείωση εμβέλειας/ποιότητας του έργου
 - (κατόπιν συζητήσεων και συμφωνίας με τον πελάτη)



Έχουμε καθυστέρηση (Behind schedule)?

- Όταν μια δραστηριότητα του κρίσιμου μονοπατιού έχει καθυστέρηση, τότε μπορεί να είναι αδύνατο να καλύψουμε το χαμένο χρόνο
- Σε αυτήν την περίπτωση ο project manager έχει 2 επιλογές:
 - **A: παράταση (να μετακινήσει μπροστά την ημερομηνία παράδοσης του έργου)**
 - **B: να μειώσει την εμβέλεια/ποιότητα του συστήματος**
- Η απόφαση πρέπει να ληφθεί σε συνεννόηση με τον πελάτη
 - Η επιλογή B επίσης απαιτεί επιπλέον ανάλυση προκειμένου να εντοπιστεί εκείνη η λειτουργία που αν αφαιρεθεί θα επιτρέψει την περάτωση του έργου στον προϋπολογισμένο χρόνο.



Διαχείριση Επαναλήψεων (Managing iteration)



Διαχείριση Επαναλήψεων (Managing iteration)

Πρόκληση: **Πώς να ελέγξουμε τον αριθμό των επαναλήψεων;**

- Υπενθύμιση: Η πρωτοτυποποίηση είναι μια αυξητική και επαναληπτική διαδικασία ανάπτυξης (*θυμηθείτε τα «εν δυνάμει μειονεκτήματα» των εξελικτικών μεθοδολογιών*)

- Οι ενέργειες πρωτοτυποποίησης πρέπει να συνοδεύονται από στόχους (που έχουν τεθεί από πριν) οι οποίοι μας επιτρέπουν να θέσουμε κριτήρια για τον έλεγχο του αριθμού των επαναλήψεων
 - Στο τέλος κάθε επανάληψης το πρωτότυπο αξιολογείται βάσει των προκαθορισμένων στόχων. Στην πράξη καμιά φορά είναι δύσκολο να καθορίσουμε αν το πρωτότυπο ικανοποιεί ή όχι τους στόχους.
 - Παράδειγμα: Έστω ότι για την Διεπαφή Χρήστη (UI) έχει τεθεί ο στόχος: να χαροποιεί τους χρήστες. Όμως η χαρά ούτε ορίζεται εύκολα, ούτε και περιορίζεται. Εκ τούτου μια χρήσιμη συμβουλή είναι: Συνέχιση επαναλήψεων (που αφορούν το UI) εωσότου λιγότερες από 5 κοσμητικές (cosmetic) αλλαγές ζητηθούν από τον πελάτη.



Διαχείριση χρόνου στις: Dynamic Systems Development Methods (DSDM)

(η μέθοδος αυτή ανήκει στις εύκαμπτες μεθοδολογίες που αναφέραμε στο 4^ο μάθημα)

Ειδοποιός διαφορά (όσον αφορά τον έλεγχο της διαδικασίας):

- **Αντί να θεωρούμε τις απαιτήσεις παγιωμένες (αμετάβλητες), η μεθοδολογία DSDM σταθεροποιεί τους πόρους (πρόσωπα, χρόνος) που θα αφιερωθούν για το έργο, και κατόπιν προσπαθεί να επιτύχει και να παραδώσει μόνο ό,τι μπορεί να επιτευχθεί βάσει αυτών των περιορισμών.**
– **Timeboxing** («χρονοπλαισίωση»)

Timeboxing: κύρια σημεία

- Καθορισμένη προθεσμία (Fixed deadline)
- Μειωμένη λειτουργικότητα (αν δεν γίνεται αλλιώς)
- Λιγότερα φινιρίσματα (“finishing touches”)

Παράδειγμα: Η Διεπαφή Χρήστη (UI) πρέπει να είναι έτοιμη έως τις 18 Δεκέμβρη και για την ανάπτυξη της δεν πρέπει να δαπανήσουμε περισσότερες από 60 εργατώρες.



DSDM > Timeboxing

Τα βήματα του “χρονοπλαισιώματος” (Timeboxing Steps)

- 1/ Καθορισμός ημερομηνίας παράδοσης
Η προθεσμία δεν πρέπει να είναι ανέφικτη
Πρέπει να ορίζεται από την ομάδα ανάπτυξης
- 2/ Ταξινόμηση των απαιτήσεων (λειτουργικών) βάσει της σπουδαιότητάς τους
- 3/ Κατασκευή του πυρήνα λειτουργίας του συστήματος
- 4/ Αναβολή της λειτουργικότητας που δεν έχει περατωθεί
- 5/ Παράδοση συστήματος με την πυρηνική λειτουργικότητα
- 6/ Επανάληψη βημάτων 3-5 για εκλέπτυνση και εμπλουτισμό της λειτουργικότητας

- We need to **prioritize the requirements** that will be actioned during a timebox
- **MoSCoW rules (Must Should Could Want)** for requirements



ή



Άλλα Οργανωτικά Θέματα (που ανήκουν στην διαχείριση έργου)

Outline:

Adopting Standards

Assessing and Managing Risk

Deciding a Directory structure for all project artifacts

The Project Plan Document

Selecting a CASE tool

Examples of planning-related artifacts from a real project



Πρότυπα (Standards)

- **Κίνητρο**
 - Όταν πολλά άτομα εργάζονται μαζί, τότε πολλά απλά πράγματα μπορούν να γίνουν εξαιρετικά μπερδεμένα
 - Μπορούμε να το αποφύγουμε υιοθετώντας πρότυπα που τα μέλη της ομάδας πρέπει να ακολουθούν. Συνάμα αυτό συμβάλει στην καλύτερη επικοινωνίας των μελών της ομάδας
- **Παραδείγματα**
 - **Τυπικοί κανόνες ονοματοδοσίας αρχείων**
 - **Φόρμες που για τους στόχους που έχουν επιτευχθεί (Forms indicating goals reached)**
 - **Κατευθυντήριες Οδηγίες για τους Προγραμματιστικές (Programming guidelines)**
 - **Καλούπια (Templates) για:**
 - Τις ατζέντες και τα πρακτικά (minutes) των συναντήσεων
 - Τα αποτελέσματα των αξιολογήσεων του έργου (project reviews)



Παραδείγματα Προτύπων

- Πρότυπα για την Τεκμηρίωση (Documentation)
 - The project name and date should appear in the header
 - Every document should have table of contents
 - Every document should be based on a agreed template
 - Agenda.dot, minutes.dot, projectReview.dot, testLog.dot, requirementsDocument.dot
- Οδηγίες προς τους προγραμματιστές
 - Each source file should have a header that lists the programmer, the last date of update and the purpose of the code
 - Indentation should indicate loops, if-then-else, case statements, ..
 - Every program should have at least one comment every 5 lines of code
- Πρότυπα Διαδικασίας
 - Every Friday at 14:00 the progress of each task should be reported
 - The project manager should approve every change in the requirements document



header.java

```
/**
 * $<remove me>Id$
 *
 * Copyright by FASTAXON-project group
 *
 * This code is licenced under VTT Public licence
 * See http://opensource.erve.vtt.fi/LICENSE.txt
 *
 * Description:
 * Author:
 * Last Update:
 */

//In order to get CVS keyword substitution to work remove <remove me>
//from above
```



Code comments > Guidelines for writing comments for the javadoc tool

`@param` (classes, interfaces, methods and constructors only)
`@return` (methods only)
`@exception` (`@throws` is a synonym added in Javadoc 1.2)
`@author` (classes and interfaces only, required)
`@version` (classes and interfaces only, required.)
`@see`
`@since`
`@serial` (or `@serialField` or `@serialData`)
`@deprecated` (see How and When To Deprecate APIs)

<http://java.sun.com/j2se/javadoc/writingdoccomments/>



Code comments

```
/**
 * $Id: javadoc.java,v 1.1 2003/10/14 20:48:26 htornroo
 * Exp $
 *
 * Copyright by FASTAXON project group
 *
 * This code is licenced under VTT Public licence
 * See http://opensource.erve.vtt.fi/LICENSE.txt
 */

//Class and Interface Documentation Tags
/**
 * A class representing a window on the screen.
 * For example:
 * <pre>
 *   Window win = new Window(parent);
 *   win.show();
 * </pre>
 *
 * @author Sami Shaio
 * @version %I%, %G%
 * @see java.awt.BaseWindow
 * @see java.awt.Button
 */
class Window extends BaseWindow {
    ...
}
```

```
//Constructor and Method Documentation Tags
/**
 * Returns the character at the specified index. An index
 * ranges from <code>0</code> to <code>length() -
 * 1</code>.
 *
 * @param index the index of the desired character.
 * @return the desired character.
 * @exception StringIndexOutOfBoundsException
 *         if the index is not in the range
 *         <code>0</code>
 *         to <code>length()-1</code>.
 * @see java.lang.Character#charValue()
 */
public char charAt(int index) {
    ...
}

//Field Documentation Tags
/**
 * The X-coordinate of the component.
 *
 * @see #getLocation()
 */
int x = 1263732;
```



Διαχείριση Κινδύνου (Risk Management)



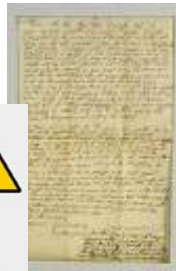
Διαχείριση Κινδύνου (Risk Management)

Είναι η διαδικασία εκτίμησης και αντιμετώπισης των κινδύνων που σχετίζονται με την ανάπτυξη ενός έργου.

- **Απαιτήσεων**
 - Αποφυγή του μεγάλου κινδύνου: της κατασκευής «λάθους συστήματος», ενός συστήματος που δεν ικανοποιεί τους πελάτες
- **Τεχνολογικοί**
 - Θα δουλέψει η επιλεγθείσα τεχνολογία;
 - Θα δέσουν τα διάφορα εξαρτήματα μεταξύ τους?
- **Ικανοτήτων (Skills)**
 - Θα βρω το προσωπικό και με τα προσόντα που απαιτούνται;
- **Πολιτικοί**
 - Υπάρχουν «πολιτικές» δυνάμεις που μπορούν να μπουν στη μέση και να επηρεάσουν σοβαρά την εξέλιξη ενός έργου?



Έγγραφο Αποτίμησης Ρίσκου (Risk Assessment Document)




Έγγραφο Αποτίμησης Ρίσκου (Risk Assessment Document)



Η ομάδα δημιουργεί (και διατηρεί ενημερωμένο) ένα έγγραφο που:

- Απαριθμεί όλα τα σχετιζόμενα ρίσκα
- Εκτιμά την πιθανότητα του κάθε ρίσκου
- Προβλέπει τις εν δυνάμει συνέπειες του κάθε ρίσκου στο έργο
- Προτείνει δράσεις για μείωση και αντιμετώπιση του ρίσκου
– (σύντομη περιγραφή, π.χ. μια παράγραφος)




Έγγραφο Αποτίμησης Ρίσκου: Παράδειγμα (1/2) (για την 1^η επανάληψη ενός πολυφασικού έργου)

T-76.633				Probabilites:	Loss impacts:	Priorities:		
risk register				1. Highly probable	1. Very high	1. Critical		
Name of person filling in the data		Tero Leppänen		2. Probable	2. High	2. Major		
Name of group:		Muuntaja		3. Possible	3. Medium	3. Medium		
Project:		Fastaxon 11		4. Unlikely	4. Low	4. Minor		
				5. Highly unlikely	5. Very low	5. Trivial		

ID	Area	Risk Factors	Risk Event	Risk Effects	Owner	Probability	Loss impact	Priority	Controlling actions	Action Status
1	Project management	1. The group is working while studying. 2. Some group members have family with small children.	1-2 persons leave the project.	1. Too much work for rest of the group. 2. Some parts of the project become unknown to group.	Project manager	5	2	4	1. Project manager will monitor the workload for individual members. 2. Assign backup person for every task.	Ongoing
2	Schedule	1. Project schedule is fixed by course 2. The group is working while studying	Project deadlines are exceeded	The group fail to pass the course	Risk management board	4	1	4	1. Assign responsible person for handling the course deliveries. 2. Set internal milestones inside iterations.	Done
3	Technical skills	The group has limited knowledge of Java	1. Coding is slow 2. Quality of code is poor	The group fails to meet the customer goals	Project manager	2	2	2	1. Plan schedule so that there is enough time to study the new tools. 2. Arrange learning sessions between head architect and rest of the group 3. Quality manager monitors SW quality	Done

U. of Crete, Information Systems Analysis and Design Yannis Tzitzikas, Fall 2006 83



Έγγραφο Αποτίμησης Ρίσκου: Παράδειγμα (2/2) (για την 1^η επανάληψη ενός πολυφασικού έργου)

4	FT theory	1. Theory behind FT is new and difficult to understand	1. Requirements are not clear. 2. Group fail to prioritize features	The group fails to meet the customer goals. FT software failed to build.	Account manager, Project manager	3	2	3	1. Monitor progress of the project. 2. Arrange reviews and meetings between customer and group	Ongoing
5	Tools	New tools and techniques	Lots of time is spend for learning tools and evaluating techniques	Important tasks are delayed. Project does not meet schedule	Project manager	3	3	3	1. Try to test whole repertoire of tools during I1 2. One person is evaluating tool first and he prepares instructions to others 3. Try to use 'standard' techniques	Done/ongoing with new tools
6	Tools	Tool compatibility problems	Lots of time is spend for tool compatibility problems	Important tasks are delayed. Project does not meet schedule	Project manager	2	3	3	1. One person test tools first and he prepares instructions to others 2. Do not use too exotic tools 3. Try to use 'standard' techniques	Done/ongoing with new tools
7	Architecture	Architecture selection effects to GUI development more than expected	1. User interface requirements are not fulfilled 2. Too much time spend to UI development	1. Schedule slips 2. Architecture change or GUI requirement change	Project manager	3	3	3	GUI design will be verified as soon as possible, latest in the beginning of the I2	Done
8	Architecture	Architecture is not well understood by all group members.	1. Much time spent on waiting other peoples results. 2. Rework	1. Workload does not deviate evenly. 2. Schedule slips	Head architect	2	2	2	1. Arrange internal workflow and architecture training session. 2. Tasks managed in a way that minimizes dependencies.	Done I1, Ongoing with I2, I3 and DE
9	Architecture	Customer has some strong opinions of technical issues	Customer steers too much technical selections	1. Architecture selection is not optimal 2. GUI is poor	Account manager	4	4	5	1. Make early releases for customer to show how selections affect 2. Negotiate with customer. Prepare arguments.	Done
10	Project management	Inter-task dependencies are not clear.	1. Much time spent on waiting other peoples results. 2. Rework	1. Schedule slips 2. Workload does not deviate evenly	Project manager	2	2	2	Iteration planning is done with co-operation with Head architect.	Done with I1 and I2. Ongoing with I3 and DE

U. of Crete, Information Systems Analysis and Design Yannis Tzitzikas, Fall 2006 84




Το έγγραφο αυτό πρέπει να ενημερώνεται

Έγγραφο Αποτίμησης Ρίσκου: Παράδειγμα (1/2)
(για την 2^η επανάληψη ενός πολυφασικού έργου)

T-76.633											
risk register						Probabilites:		Loss impacts:		Priorities:	
Name of person filling in the dat		Hannu Törnroos				1. Highly probable		1. Very high		1. Critical	
Name of group:		Muuntaja				2. Probable		2. High		2. Major	
Project:		Fastaxon I2				3. Possible		3. Medium		3. Medium	
						4. Unlikely		4. Low		4. Minor	
						5. Highly unlikely		5. Very low		5. Trivial	
ID	Area	Risk Factors	Risk Event	Risk Effects	Owner	Probability	Loss impact	Priority	Controlling actions	Action Status	
1	Project management	1. Important work cannot be divided to small tasks 2. Work is divided unevenly	Architect and GUI responsible can't handle the workload	1. These persons have to make extra hours (>190) 2. Schedule slips due the waiting of their work.	Project manager	2	2	2	1. Make sure architect and GUI responsible do only tasks that can't be done by anyone else 2. Use Yannis in coding to reduce architects workload.	Ongoing	
2	Schedule	1. Project schedule is fixed by course 2. The group is working while studying	Project deadlines are exceeded	Iteration grade drops due missed DL	Project manager	4	2	4	1. Assign responsible person for handling the course deliveries. 2. Set internal milestones inside iterations.	Ongoing	
3	Dev/test environment	Dev/test environment installation is hard	Much time spent in setting up new/broken environments	1. Schedule slips. 2. Developers testers will become frustrated work morale drops	Test Manager	3	3	3	1. Fix installation instructions 2. Gather instructions in one manual.	Assigned to Pekka Uusitalo	
4	FT theory	1. Theory behind FT is new and difficult to understand	1. Requirements are not clear. 2. Group fail to prioritize	The group fails to meet the customer goals. FT software failed to build.	Account manager, Project manager	3	2	3	1. Monitor progress of the project. 2. Arrange reviews and meetings between customer and group	Ongoing	

U. of Crete, Information Systems Features and Design Yannis Tzitzikas, Fall 2006 86

 Έγγραφο Αποτίμησης Ρίσκου: Παράδειγμα (2/2) (για την 2^η επανάληψη ενός πολυφασικού έργου)										
5	Software quality	1. Requirements document does not offer enough data for writing test cases 2. Testing is apart from rest of the project	Testers are testing obsolete/wrong functionality.	Software quality is bad (or quality is in wrong places)	Test Manager	3	2	3	1. Try to read reqs. data "between the lines" 2. Arrange meeting to clarify requirements in testing point of view	Done/ongoing with 13
6	Project management	Requirements are in too abstract level	Coders misunderstand reqs	Important tasks are delayed. Project does not meet schedule	Project manager	2	2	2	1. If something is unclear ask Mika immediately. 2. If Mika does not know the answer ask Yannis immediately. 3. Project manager to cc:-field of all FX emails.	Ongoing
7	Requirements	Project group did not know the substance well enough during reqs. elicitation	Important reqs were missed during reqs elicitation	1. Rework to already implemented functionality 2. Some requirements have to be dropped to have time for new ones.	Project manager	3	2	3	1. Strict discipline in change management 2. Follow the SCM process defined in project plan.	Ongoing
8	Architecture	Architecture is not well understood by all group members.	1. Much time spent on waiting other peoples results. 2. Rework	1. Workload does not deviate evenly. 2. Schedule slips	Head architect	2	2	2	1. Arrange Pair programming session 2. Tasks managed in a way that minimizes dependencies.	Pair programming sessions arranged by both GUI&mdel coding "teams". Task management ongoing with 13 and DE
9	Requirements	1. New requirements elicited during work 2. Changes are not officially approved by customer	Unplanned functionality implemented instead of planned	1. Less time for the important stuff 2. Unplanned functionality won't be documented (and thus won't get tested)	change management board	1	3	2	1. Strict discipline in change management 2. Prioritize new reqs with customer before implementation	Ongoing
10	FX theory	1. Navigation tree and CheckValidity algorithms are hard to understand	The most important part of SW does not work	1. System is useless	Architect	3	1	1	1. Prioritize algorithm work ahead of everything else 2. Use Yannis in	Ongoing



Αποφασίζοντας τη Δομή των Φακέλων



Παράδειγμα (1/2)

```

projectName/
  development/
    source/
      lib/
      lib_x_project/
      storage_manager/
        project_x/
        project_y/
      expression_builder/
        project_x/
        project_y/
    share/
  database/
    design/ (DB design etc.)
    content/ (DB contents for testing)
  testing/
    test cases/
    test diaries/
    test logs/

```

```

bin/
  release/
    rel_1_0_0/
      doc/
      client/
        bin/
        src/
        doc/
      server/
        bin/
        src/
        doc/
      database/
        design/
        content/

```



Παράδειγμα (2/2)

```

doc/
  specifications/
    requirement specifications/
    design specifications/
    architectural specifications/
  templates/
    src/
      IDE project templates
      src file templates
    doc/
      minutes template
      etc...
  development environment specifications/
  development HW.txt (?)
  operating systems.txt (?)
  development SW.txt (?)
  dev env installation manual.txt

```

```

development instructions and standards/
  developers guide.txt (?)
  testers guide.txt (?)
  followed standards.txt (?)
manuals/
  installation manuals/ (?)
  user manuals/ (?)
quality assurance/
  inspection materials
  checklists/
  minutes/
  reports/
change management/ (?)
  error reports/ (?)
  change requests/ (?)
administrative documents/
  meeting minutes/
  iteration plans/ (project plans)
  maintenance plans/ (?)
  quality assurance plan/ (?)
  configuration management plan/ (?)
  agreements/
marketing materials/ (?)

```



Έγγραφο Πλάνου Έργου (Project Plan Document)



Έγγραφο Πλάνου Έργου (Project Plan Document): Μια τυπική διάρθρωση (1/5)



Για ένα έργο που θα αναπτυχθεί πολυφασικά

1. Εισαγωγή
2. Δικαιούχοι και Στελέχωση
3. Στόχοι και κριτήρια τερματισμού
4. Πόροι και Προϋπολογισμός
5. Πρακτικές και Εργαλεία
6. Φάσεις
7. Πλάνο Διαχείρισης Κινδύνου



Έγγραφο Πλάνου Έργου (Project Plan Document): Μια τυπική διάρθρωση (2/5)

1. Εισαγωγή
 - Σκοπός και εμβέλεια του έργου
 - Το σύστημα και το περιβάλλον του
 - Δικαιώματα (πνευματικά/εκμετάλλευσης) των Προϊόντων του Έργου
 - Ορολογία και Ορισμοί
2. Δικαιούχοι και Στελέχωση
 - Ομάδα έργου (ρόλοι)
 - Άλλοι δικαιούχοι
3. Στόχοι και κριτήρια τερματισμού
4. Πόροι και Προϋπολογισμός
5. Πρακτικές και Εργαλεία
6. Φάσεις
7. Πλάνο Διαχείρισης Κινδύνου



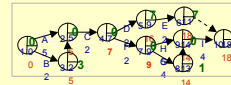
Έγγραφο Πλάνου Έργου (Project Plan Document): Μια τυπική διάρθρωση (3/5)

1. Εισαγωγή
2. Δικαιούχοι και Στελέχωση
3. Στόχοι και κριτήρια τερματισμού
 - Στόχοι του Πελάτη
 - Βασικές (core) λειτουργίες και χαρακτηριστικά του συστήματος, κορυφαίοι 10 στόχοι (top-10)
 - Πρακτικές εργασίας, χρονοδιάγραμμα, τεκμηρίωση
 - Κριτήρια Επικύρωσης (verification)
 - Αντικειμενικά (κατά προτίμηση μετρήσιμα)
 - Υποκειμενικά (customer evaluates ...)
 - Στόχοι της Ομάδας Έργου
 - Κριτήρια Διακοπής του Έργου
 - Κριτήρια Περάτωσης του Έργου
4. Πόροι και Προϋπολογισμός
5. Πρακτικές και Εργαλεία
6. Φάσεις
7. Πλάνο Διαχείρισης Κινδύνου



Έγγραφο Πλάνου Έργου (Project Plan Document): Μια τυπική διάρθρωση (4/5)

1. Εισαγωγή
2. Δικαιούχοι και Στελέχωση
3. Στόχοι και κριτήρια τερματισμού
4. Πόροι και Προϋπολογισμός
 - Προσωπικό
 - Διαθέσιμη Υποδομή (hardware/ software resources)
 - Προϋπολογισμός (χρηματική)
5. Πρακτικές και Εργαλεία
 - Πρακτικές
 - Εργαλεία
 - Πρότυπα
6. Φάσεις
 - χρονοδιάγραμμα, ορόσημα (milestones), πλάνο επαναλήψεων (στόχοι, παραδοτέα, εργασίες)
7. Πλάνο Διαχείρισης Κινδύνου



Έγγραφο Πλάνου Έργου (Project Plan Document): Μια τυπική διάρθρωση (5/5)

1. Εισαγωγή
2. Δικαιούχοι και Στελέχωση
3. Στόχοι και κριτήρια τερματισμού
4. Πόροι και Προϋπολογισμός
5. Πρακτικές και Εργαλεία
6. Φάσεις
7. Πλάνο Διαχείρισης Κινδύνου
 - Πρακτικές Διαχείρισης Ρίσκου
 - Ποιος είναι υπεύθυνος για τη Διαχ. Ρίσκου?
 - Πως θα γίνεται η διαχείριση?
 - Πόσο συχνά και από ποιόν θα αναθεωρείται το σχετικό έγγραφο
 - Κίνδυνοι
 - ρίσκο, συνέπειες/επιπτώσεις (effects) , δράσεις αντιμετώπισης (controlling actions)



Συνηθισμένα Λάθη



Κλασσικά Λάθη Διαχείρισης Έργου

- Υπεραισιόδοξο χρονοδιάγραμμα
- Μη παρακολούθηση χρονοδιαγράμματος
- Μη ενημέρωση χρονοδιαγράμματος
- Προσθήκη ανθρώπων σε ένα καθυστερημένο έργο



Εργαλεία για Διαχείριση Έργου



Εργαλεία για Διαχείριση Έργου

Υπάρχουν δεκάδες διαθέσιμα εργαλείων, π.χ. δείτε

- http://en.wikipedia.org/wiki/Project_management_software
- <http://www.startwright.com/project1.htm>

Μερικά συχνά χρησιμοποιούμενα εργαλεία:

- Microsoft VISIO και Microsoft Project
 - Σημείωση: Για τα προγράμματα της Microsoft έχουμε άδεια ακαδημαϊκού ιδρύματος άρα μπορείτε να τα πάρετε και να τα χρησιμοποιείτε στο πανεπιστήμιο και στο σπίτι σας. Για περισσότερες πληροφορίες ρωτήστε τους βοηθούς ή τους διαχειριστές (administrators).



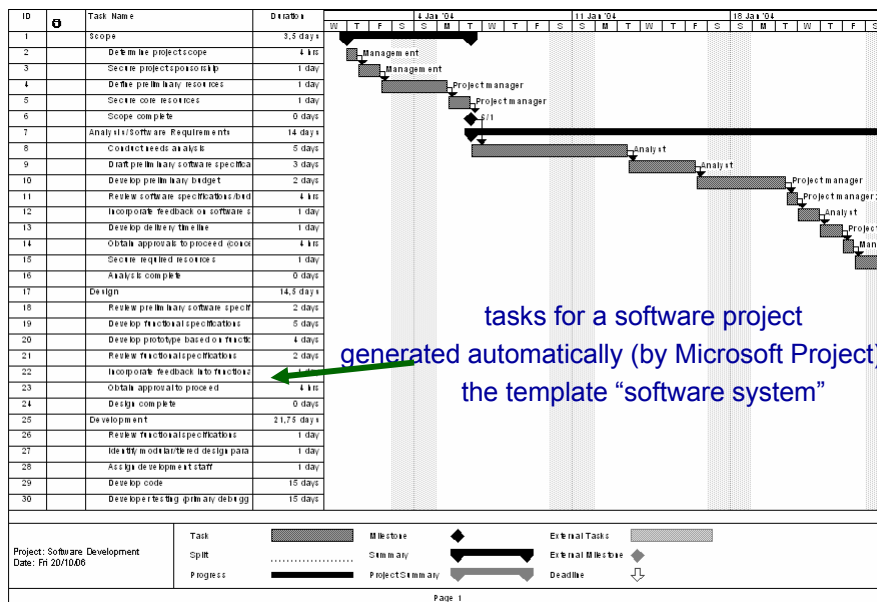
... Microsoft Project

• Tutorials:

- <http://www.urban.uiuc.edu/Courses/varkki/msProject/msproject.html>
- <http://greenbay.usc.edu/csci577/spring2006/site/tools/Project/MSProjectTutorial.ppt>



... Microsoft Project > Gantt charts





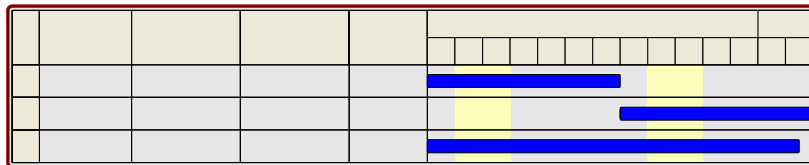
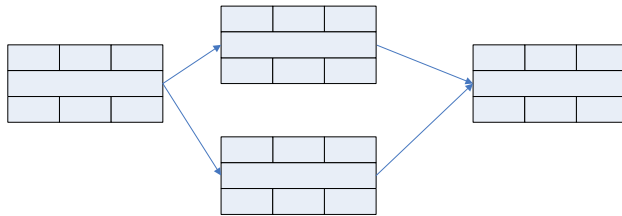
... Microsoft Project > View > task usage

ID	Task Name	Work	Details	1 Jan '04							11 Jan '04								
				T	F	S	S	M	T	W	T	F	S	S	M	T	W		
1	Scope	28 hr	Work																
2	Define project scope	4 hr	Work																
3	Secure project resources	5 hr	Work																
4	Define preliminary budget	3 hr	Work																
5	Secure resources	3 hr	Work																
6	Scope complete	0 hr	Work																
7	Analyze Software Requirements	120 hr	Work																
8	Conduct user analysis	40 hr	Work																
9	Develop preliminary software	24 hr	Work																
10	Develop preliminary budget	16 hr	Work																
11	Review software specifications	4 hr	Work																
12	Incorporate feedback on	3 hr	Work																
13	Develop user interface	3 hr	Work																
14	Obtain approval to proceed	3 hr	Work																
15	Secure needed resources	3 hr	Work																
16	Analysis complete	0 hr	Work																
17	Design	120 hr	Work																
18	Develop preliminary software	16 hr	Work																
19	Develop functional specifications	40 hr	Work																
20	Develop prototype based	32 hr	Work																
21	Review functional specifications	16 hr	Work																
22	Incorporate feedback into	3 hr	Work																
23	Obtain approval to proceed	3 hr	Work																
24	Design complete	0 hr	Work																
25	Develop unit	24 hr	Work																
26	Review functional specifications	3 hr	Work																
27	Identify and distribute de	3 hr	Work																
28	Acceptance complete	3 hr	Work																



Συγκριζοντας με το Microsoft VISIO

- PERT Diagrams and Gantt charts





- **Object-Oriented Systems Analysis and Design Using UML** (2nd edition) by S. Bennett, S. McRobb, R. Farmer, McGraw Hill, 2002. CHAPTER 21
- **Systems Analysis and Design with UML Version 2.0** (2nd edition) by A. Dennis, B. Haley Wixom, D. Tegarden, Wiley, 2005. CHAPTER 4
- **System Analysis and Design Methods** (6th edition) by Jeffrey L. Whitten, Lonnie D. Bentley and Kevin Dittman, McGraw-Hill, 2004, Chapter 4