



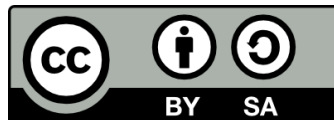
ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΑΝΑΛΥΣΗ
Rational Unified Process

Ιωάννης Σταμέλος
Βάιος Κολοφωτιάς
Πληροφορική

Άδειες Χρήσης

Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons. Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα. Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.



Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Περιεχόμενα

Άδειες Χρήσης.....	2
Χρηματοδότηση.....	2
1. Περιεχόμενο Μαθήματος.....	4
2. Περιεχόμενα ενότητας.....	4
3. ΕΙΣΑΓΩΓΗ.....	5
3.1 Ο κύκλος ζωής του λογισμικού και οι Φάσεις.....	5
3.1.1 Οι τέσσερις φάσεις είναι:.....	5
3.1.2 Ορισμοί.....	5
3.1.3 Ορόσημα (<i>milestones</i>).....	6
4. Οι Φάσεις.....	6
4.1.1 Φάση Σύλληψης.....	6
4.1.2 Φάση Επεξεργασίας.....	6
4.1.3 Φάση Κατασκευής.....	7
4.1.4 Φάση Μετάβασης.....	7
4.2 Η διαδικασία λογισμικού (<i>Software Process</i>).....	8
4.3 Ρόλοι - Δραστηριότητες - Αρχέτυπα.....	8
4.3.1 Ρόλος (<i>role</i>).....	9
4.3.2 Δραστηριότητα (<i>Activity</i>).....	9
4.3.3 Αρχέτυπο (<i>Artifact</i>).....	9
4.4 Τομέας.....	10
4.4.1 Οι τομείς.....	10
4.5 Η ένωση του κύκλου ζωής και της διεργασίας λογισμικού.....	11
4.6 Τομείς και Φάσεις.....	11
4.7 Επανάληψη.....	11

1. Περιεχόμενο Μαθήματος

Εβδομάδα	Περιεχόμενο
1 ^η	Εισαγωγή στην Αντικειμενοστρεφή Ανάλυση/UML
2 ^η	Rational Unified Process
3 ^η	Περιπτώσεις Χρήσης
4 ^η	Διαγράμματα Κλάσεων
5 ^η	Διαγράμματα Συνεργασίας
6 ^η	Διαγράμματα Ακολουθίας
7 ^η	Πρότυπα Σχεδίασης
8 ^η	Διεργασία ICONIX
9 ^η	Επιχειρηματική Μοντελοποίηση
10 ^η	Υλοποίηση Σχεδίασης με Java
11 ^η	Μετρικές Αντικειμενοστραφούς Σχεδίασης
12 ^η	Επισκόπηση

2. Περιεχόμενα ενότητας

Σε αυτή την ενότητα θα δούμε την RUP(Rational Unified Process) μια διαδικασία τεχνολογίας λογισμικού που διασφαλίζει την ποιότητα του καθόλου τον κύκλο ζωής του. Θα αναλύσουμε τις φάσεις σύλληψης, επεξεργασίας και μετάβασης και θα μάθουμε πως ορίζονται ο ρόλος, η δραστηριότητα και το αρχέτυπο στον κόσμο της RUP.

3. ΕΙΣΑΓΩΓΗ

- Η RUP είναι μια Διαδικασία Τεχνολογίας Λογισμικού (Software Engineering Process).
- Ο στόχος της είναι να διασφαλίσει την παραγωγή λογισμικού υψηλής ποιότητας που ικανοποιεί τις ανάγκες των τελικών χρηστών μέσα σε ένα συγκεκριμένο χρονοδιάγραμμα και κόστος.

3.1 Ο κύκλος ζωής του λογισμικού και οι Φάσεις

Ο κύκλος ζωής του λογισμικού (software life-cycle) σκιαγραφεί τη ζωή του προγράμματος λογισμικού από τη στιγμή της γέννησής του μέχρι τη στιγμή της αντικατάστασης ή της εγκατάλειψής του.

Ο κύκλος ζωής του λογισμικού στην RUP υποδιαιρείται σε τέσσερις συνεχόμενες φάσεις.

3.1.1 Οι τέσσερις φάσεις είναι:

- Η Φάση Σύλληψης (*Inception Phase*),
- Η Φάση Επεξεργασίας (*Elaboration Phase*),
- Η Φάση Κατασκευής (*Construction Phase*) και
- Η Φάση Μετάβασης (*Transition Phase*).

3.1.2 Ορισμοί

- **Κύκλος ανάπτυξης:** Ένα πέρασμα από τις 4 φάσεις.
- Κάθε πέρασμα από τις 4 φάσεις παράγει μια νέα **γενιά** (*generation*) λογισμικού.
- Οι διαδοχικοί κύκλοι στην περίπτωση που το προϊόν συνεχίσει να εξελίσσεται στην επόμενη γενιά του, ονομάζονται **κύκλοι εξέλιξης** (*evolution cycles*).

3.1.3 Ορόσημα (*milestones*)

Καθεμία από τις φάσεις ολοκληρώνεται με την παράδοση κάποιων ορόσημων. Στο τέλος κάθε φάσης πραγματοποιείται μια αποτίμηση η οποία κρίνει εάν οι στόχοι της φάσης έχουν ικανοποιηθεί προκειμένου να αρχίσει η επόμενη φάση.

4. Οι Φάσεις

4.1.1 Φάση Σύλληψης

Αποτελέσματα της φάσης Σύλληψης είναι τα εξής :

- Μία γενική περιγραφή των βασικών απαιτήσεων του έργου, τα σημεία κλειδιά και οι βασικοί περιορισμοί.
- Ένα αρχικό μοντέλο περιπτώσεων χρήσης (ολοκληρωμένο κατά 10%-20%)
- Ένα αρχικό γλωσσάριο του έργου.
- Μια αρχική εκτίμηση των κινδύνων.
- Μία αρχική επιχειρηματική περίπτωση, η οποία περιέχει το επαγγελματικό πλαίσιο, κριτήρια επιτυχίας και οικονομική πρόβλεψη.
- Ένα σχέδιο του έργου (project plan), το οποίο δείχνει φάσεις και επαναλήψεις.
- Ένα επιχειρηματικό μοντέλο (business model) αν είναι απαραίτητο.
- Ένα ή διάφορα πρωτότυπα

4.1.2 Φάση Επεξεργασίας

Το αποτέλεσμα της φάσης επεξεργασίας είναι:

- Ένα μοντέλο περιπτώσεων χρήσης (ολοκληρωμένο τουλάχιστον κατά 80%)
- Συμπληρωματικές απαιτήσεις για τις μη λειτουργικές απαιτήσεις και απαιτήσεις που δεν έχουν σχέση με μια συγκεκριμένη περίπτωση χρήσης.
- Ένα εκτελέσιμο αρχιτεκτονικό πρωτότυπο
- Μια περιγραφή της αρχιτεκτονικής του λογισμικού
- Επανεξέταση της λίστας κινδύνων και της επιχειρηματικής περίπτωσης.
- Ένα σχέδιο ανάπτυξης για το ολικό έργο
- *Το έργο μπορεί να ματαιωθεί ή να επανεξεταστεί σοβαρά η πραγματοποίησή του αν αποτύχει να περάσει από αυτό το ορόσημο*

4.1.3 Φάση Κατασκευής

Αποτελείται τουλάχιστον από τα εξής:

- Το προϊόν λογισμικού
- Το εγχειρίδιο του χρήστη
- Μια περιγραφή της παρούσας άδειας έκδοσης

Η μετάβαση μπορεί να αναβληθεί αν το έργο αποτύχει να περάσει αυτό το ορόσημο.

4.1.4 Φάση Μετάβασης

Ο σκοπός της φάσης είναι η μετάβαση του λογισμικού προϊόντος στην κοινότητα των χρηστών. Αφού το προϊόν παραδοθεί στους τελικούς χρήστες, συνήθως εντοπίζονται προβλήματα που απαιτούν την ανάπτυξη νέων εκδόσεων.

Σε αυτό το σημείο αποφασίζεται αν οι στόχοι έχουν επιτευχθεί και αν θα πρέπει να ξεκινήσει ο επόμενος κύκλος εξέλιξης.

4.2 Η διαδικασία λογισμικού (*Software Process*)

- παρέχει τις κατευθυντήριες γραμμές για την αποτελεσματική ανάπτυξη και εξέλιξη του λογισμικού.
- θέτει τα θεμέλια για τις επόμενες γενιές του προϊόντος,
- καθορίζει ποιος κάνει τι, πότε και πώς να φτάσει στο στόχο. Κάτι τέτοιο αυξάνει την επίδοση των δημιουργών λογισμικού.
- περιέχει όλες τις δραστηριότητες που απαιτούνται για να τροποποιηθούν οι ανάγκες του χρήστη σε προϊόν λογισμικού.
- Η διαδικασία λογισμικού (*Software Process*) (1)
- παρέχει τις κατευθυντήριες γραμμές για την αποτελεσματική ανάπτυξη και εξέλιξη του λογισμικού.
- θέτει τα θεμέλια για τις επόμενες γενιές του προϊόντος,
- καθορίζει ποιος κάνει τι, πότε και πώς να φτάσει στο στόχο. Κάτι τέτοιο αυξάνει την επίδοση των δημιουργών λογισμικού.
- περιέχει όλες τις δραστηριότητες που απαιτούνται για να τροποποιηθούν οι ανάγκες του χρήστη σε προϊόν λογισμικού.

4.3 Ρόλοι - Δραστηριότητες - Αρχέτυπα

- Η RUP διαδικασία λογισμικού αποτελείται από τομείς, οι οποίοι καθορίζουν ένα σύνολο **δραστηριοτήτων** που εκτελούνται από τους **ρόλους** για τη δημιουργία των αρχετύπων.
- Έτσι λοιπόν η RUP βασίζεται στην ιδέα ότι η διαδικασία λογισμικού είναι μια συνεργασία ανάμεσα σε αφηρημένες ενεργές οντότητες που ονομάζονται ρόλοι, οι οποίοι εκτελούν λειτουργίες που ονομάζονται δραστηριότητες πάνω σε διακριτές οντότητες που ονομάζονται αρχέτυπα.

4.3.1 Ρόλος (role)

- Ο όρος **ρόλος** καθορίζει τις ευθύνες των ανθρώπων που φέρνουν σε πέρας τις δραστηριότητες της διαδικασίας λογισμικού.
- Ο ρόλος ενός ατόμου εξελίσσεται στη διάρκεια του χρόνου και εξαρτάται από τις δραστηριότητες που πρέπει να εκτελεστούν το συγκεκριμένο χρόνο.

4.3.2 Δραστηριότητα (Activity)

- Μια δραστηριότητα είναι η μικρότερη δυνατή εργασία. Τεχνικά είναι μια λειτουργία που εκτελείται από ένα άτομο με ένα συγκεκριμένο ρόλο.
- Μια δραστηριότητα πρέπει να έχει ένα ξεκάθαρο σκοπό και συνήθως πρόκειται για τη δημιουργία ή ανανέωση αρχετύπων όπως ένα μοντέλο ή μια κλάση
- Η διάρκεια μιας δραστηριότητας κυμαίνεται από μερικές ώρες σε μερικές μέρες.

4.3.3 Αρχέτυπο (Artifact)

- Ένα αρχέτυπο είναι ένα κομμάτι πληροφορίας το οποίο παράγεται, τροποποιείται ή χρησιμοποιείται από τη διαδικασία λογισμικού.
- Χρησιμοποιούνται σε μια δραστηριότητα και ταυτόχρονα είναι το αποτέλεσμα της δραστηριότητας. Τέλος ένα αρχέτυπο μπορεί να αποτελείται από άλλα αρχέτυπα, όπως το μοντέλο σχεδίασης αποτελείται από πολλές κλάσεις.

4.4 Τομέας

- Η καταγραφή των ρόλων, των δραστηριοτήτων και των αρχετύπων δεν είναι επαρκής για να αποτελέσει μια διαδικασία.
- Η ενοποίηση αυτών των τριών οντοτήτων έχει ως αποτέλεσμα μια μεγάλη οντότητα η οποία αναφέρεται ως τομέας, ενώ μια διαδικασία είναι μια ομάδα από τομείς.

4.4.1 Οι τομείς

- **Ο τομέας απαιτήσεων** (*requirements discipline*) καθορίζει ένα μοντέλο περιπτώσεων χρήσης για να περιγράψει τη συμπεριφορά του συστήματος.
- **Ο τομέας ανάλυσης και σχεδίασης** (*analysis and design discipline*) έχει ως αποτέλεσμα τη δημιουργία ενός μοντέλου σχεδίασης, το οποίο μεταφράζει τις απαιτήσεις σε κάτι πιο συγκεκριμένο το οποίο περιγράφει πως θα υλοποιηθεί το σύστημα.
- **Ο τομέας υλοποίησης** (*implementation discipline*) είναι υπεύθυνος για την οργάνωση του κώδικα, υλοποιεί τις κλάσεις και τα αντικείμενα, ελέγχει τις δημιουργημένες συνιστώσες και ενώνει τα διάφορα τμήματα σε ένα εκτελέσιμο σύστημα.
- **Ο τομέας ελέγχου** (*testing discipline*) εκτιμά ποιό είναι το επίπεδο ποιότητας που επιτεύχθηκε στο προϊόν και αναφέρει τα αποτελέσματα.
- **Ο τομέας διαμόρφωσης και αλλαγής διοίκησης** (*configuration and change management discipline*) Αυτός ο τομέας μπορεί να ελέγξει αλλαγές, να αναφέρει, να καταγράψει την κατάσταση των αρχέτυπων και τις απαιτήσεις για τροποποίηση, να επιβεβαιώσει την ολοκλήρωση, τη συνέπεια και την ορθότητα των αρχέτυπων και τέλος να ελέγξει την αποθήκευση, τη μεταχείριση και την παράδοση τους.
- **Ο τομέας διοίκησης έργου** (*project management discipline*) εξισορροπεί συγκρουόμενους στόχους, διαχειρίζεται τους κινδύνους και ξεπερνά τους περιορισμούς που εμφανίζονται. Αυτός ο τομέας επικεντρώνεται στο χρόνο, στο κόστος, στους ανθρώπους, στη διαδικασία και σε άλλες οικονομικού ενδιαφέροντος προτάσεις.

4.5 Η ένωση του κύκλου ζωής και της διεργασίας λογισμικού

- Από τη μια πλευρά, ο κύκλος ζωής του λογισμικού αποτελείται από φάσεις μέσω των οποίων περνάει το προϊόν λογισμικού.
- Από την άλλη πλευρά η διαδικασία λογισμικού αποτελείται από τομείς από τους οποίους πρέπει να περάσουν οι δημιουργοί λογισμικού για να ολοκληρώσουν το προϊόν.
- **Μια από τις βασικές ιδέες της RUP είναι ότι όλες οι φάσεις παίρνουν μέρος σε κάθε κύκλο εξέλιξης ή ανάπτυξης και ότι κάθε τομέας εφαρμόζεται σε κάθε φάση του κύκλου ζωής.**

4.6 Τομείς και Φάσεις

- Σε κάθε φάση του κύκλου ανάπτυξης, ένας τομέας είναι πιο σημαντικός από έναν άλλο. Για παράδειγμα, είναι επόμενο ότι ο τομέας των απαιτήσεων θα είναι πιο σημαντικός στη φάση σύλληψης και επεξεργασίας σε σύγκριση με τις επόμενες φάσεις.
- Η ιδέα είναι ότι κάθε τομέας συνεισφέρει, σε κάποιο βαθμό, σε κάθε φάση του κύκλου ζωής.
- Αυτή η ιδέα μπορεί να αναπαρασταθεί σε ένα δισδιάστατο μοντέλο της διαδικασίας και του κύκλου ζωής και κάθε στήλη, ή επανάληψη (iteration), αναφέρεται στην εφαρμογή όλων των τομέων σε μια φάση του κύκλου ζωής

4.7 Επανάληψη

Οι επαναλήψεις είναι δομές διαχείρισης που κάνουν δυνατό τον έλεγχο της ολοκλήρωσης κάθε φάσης, χρησιμοποιώντας τις κατάλληλες δραστηριότητες και αρχέτυπα που αποτελούν τον τομέα.

Κάθε φάση του κύκλου ζωής μπορεί να αποτελείται από μία ή περισσότερες επαναλήψεις. Αλλά σε κάθε επόμενη επανάληψη και φάση η έμφαση στις διάφορες δραστηριότητες του τομέα αλλάζει.