Οδηγός εφαρμογής MSF

Aris Gounaras

Πίνακας Περιεχομένων

ΕΙΣΑΓΩΓΗ **3**

ΟΜΑΔΑ ΕΡΓΟΥ 3
ΠΙΝΑΚΑΣ ΠΡΟΣΑΡΜΟΓΗΣ (TRADEOFF MATRIX) 4
ΕΠΑΝΑΛΗΨΕΙΣ ΚΑΙ ΕΚΔΟΣΕΙΣ (ITERATION & VERSIONED RELEASES) 4

ΟΡΑΜΑΤΙΣΜΟΣ (ENVISIONING PHASE) **4**

ΈΓΓΡΑΦΑ ΕΡΓΑΣΙΑΣ 4
ΚΟΜΒΙΚΑ ΣΗΜΕΙΑ 5
ΠΑΡΑΔΟΤΕΑ 5
Το έγγραφο «Σχέδιο του έργου» *5*
Το έγγραφο «Σκοπός & εύρος έργου» *6*
Το έγγραφο «Ανάλυση κινδύνων» *6*

ΑΝΑΛΥΣΗ (PLANNING PHASE) **7**

ΈΓΓΡΑΦΑ ΕΡΓΑΣΙΑΣ 7
ΚΟΜΒΙΚΑ ΣΗΜΕΙΑ 7
ΠΑΡΑΔΟΤΕΑ 7
ΕΝΝΟΙΟΛΟΓΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ 7
Επιλογή αρχιτεκτονικής *8*
Εξειδίκευση απαιτήσεων *8*
Εξειδίκευση use cases *9*
Δημιουργία σεναρίων χρήσης (usage scenarios) *9*
Έλεγχος και βελτιστοποίηση του εννοιολογικού σχεδιασμού *9*
ΛΟΓΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ 10
Λογικό μοντέλο αντικειμένων (logical object model) *10*
Έλεγχος του λογικού σχεδιασμού *11*
ΦΥΣΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ 11
UML modeling *11*
Περιβάλλον χρήστη (presentation layer) *11*
Διαχείριση δεδομένων (data layer) *11*
Προδιαγραφές ασφαλείας (security specifications) *12*
Ανάλυση απειλών (threat modeling) *12*
Άλλα σημεία προσοχής *13*
ΠΡΟΕΤΟΙΜΑΣΙΑ ΤΩΝ ΕΠΟΜΕΝΩΝ ΣΤΑΔΙΩΝ 13
Προετοιμασία για το στάδιο της ανάπτυξης *13*
Προετοιμασία για το στάδιο της σταθεροποίησης *13*
Προετοιμασία για το στάδιο της εγκατάστασης *14*
Τεχνικές προδιαγραφές *14*
ΤΟ ΕΓΓΡΑΦΟ «ΣΥΝΟΨΗ ΛΕΙΤΟΥΡΓΙΚΩΝ ΠΡΟΔΙΑΓΡΑΦΩΝ» 14
ΤΟ ΕΓΓΡΑΦΟ «ΠΛΑΝΟ ΚΑΙ ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ ΤΟΥ ΕΡΓΟΥ» 14
ΤΟ ΕΓΓΡΑΦΟ «ΣΧΕΔΙΟ ΑΣΦΑΛΕΙΑΣ» 14

ΑΝΑΠΤΥΞΗ (DEVELOPING PHASE) **15**

ΣΤΑΘΕΡΟΠΟΙΗΣΗ (STABILIZING PHASE) **15**

ΚΟΜΒΙΚΑ ΣΗΜΕΙΑ 15
ΠΑΡΑΔΟΤΕΑ 15

ΕΓΚΑΤΑΣΤΑΣΗ (DEPLOYING PHASE) **15**

ΠΡΑΚΤΙΚΕΣ ΠΑΡΑΤΗΡΗΣΕΙΣ **16**

ΕΠΙΛΟΓΟΣ **16**

Εισαγωγή

Το παρόν αποτελεί ένα συνοπτικό οδηγό εφαρμογής της μεθοδολογίας MSF. Βασίζεται στο "Analyzing requirements and defining Microsoft .Net solution architectures"[1]. Περιλαμβάνει επεξηγήσεις και σύντομα σχόλια (σύμφωνα με τον τρόπο που κατανοώ ορισμένα πράγματα). Επίσης, έγινε προσπάθεια να αποδοθεί η αγγλική ορολογία στα ελληνικά.

Περιγράφει τα στάδια (phases), σημειώνοντας τα κομβικά σημεία (milestones) και τα παραδοτέα (deliverables) του κάθε σταδίου. Επίσης, αναφέρεται στις υποχρεώσεις και στόχους της ομάδας έργου (team model & roles). Τέλος, θεωρώ ότι η μοντελοποίηση γίνεται χρησιμοποιώντας UML και ERD.

Τα στάδια της διαδικασίας είναι:

1. Οραματισμός (envisioning phase)
2. Ανάλυση (planning phase)
3. Ανάπτυξη (developing phase)
4. Σταθεροποίηση (stabilizing phase)
5. Εγκατάσταση (deploying phase)

Ομάδα έργου

Η ομάδα έργου αποτελείται από άτομα που εντάσσονται στους παρακάτω ρόλους:

Αντιμετωπίζουν το έργο από την σκοπιά του πελάτη

- Επαφή με πελάτη (Product management)
  Τα άτομα της ομάδας αυτής ασχολούνται με τις επαφές με τον πελάτη, την συλλογή των απαιτήσεων και την διαχείριση των προσδοκιών που έχει από το έργο και ελέγχουν ότι καλύπτονται οι επιχειρησιακές ανάγκες που οδήγησαν σε αυτό. Επίσης, ενημερώνουν το πελάτη, κάνουν επιδείξεις και προώθηση (marketing) του έργου στους χρήστες.
- Ανάλυση & ανάπτυξη (Program management)
  Υπεύθυνοι για την σχεδίαση και την παρακολούθηση της ανάπτυξης.
- Ομάδα ανάπτυξης (Development)
- Δοκιμαστές (Testing)
  Υπεύθυνοι για την πιστοποίηση της σχεδίασης του έργου (δηλαδή η σχεδίαση, όταν υλοποιηθεί θα κάνει ότι είχε προβλεφθεί) και για την πιστοποίηση της ποιότητας (με δοκιμές). Τέλος, πιστοποιούν ότι η λύση είναι έτοιμη για διανομή.

Έλεγχος ποιότητας

- Εμπειρία χρήστη (User experience)
  Αναλύουν τις απαιτήσεις απόδοσης, τα θέματα υποστήριξης χρηστών και τις επιπτώσεις που έχουν αυτά στην λύση (ώστε να καλυφθούν)
- Διαχείριση εκδόσεων (Release management)
  Ασχολούνται με την εγκατάσταση και την καθημερινή λειτουργία της λύσης. Επίσης, ελέγχουν για την ύπαρξη της απαιτούμενης τεχνικής υποδομής.

Σε μικρά έργα, κάποια μέλη της ομάδας μπορεί να εκπληρώνουν πάνω από ένα ρόλο. Σε αυτή την περίπτωση, οι τρεις πρώτοι ρόλοι μπορούν να καλύψουν και τους υπόλοιπους, όπου[2]:

- Ομάδα ανάπτυξης → δεν συνίσταται να κάνει κάτι άλλο
- Επαφή με πελάτη → Δοκιμαστές & Εμπειρία χρήστη
- Ανάλυση & Ανάπτυξη → Διαχείριση εκδόσεων

[1] Το βιβλίο του MCSD self-paced training kit, για το διαγώνισμα 70-300
[2] Ο πλήρης πίνακας συνδυασμών είναι στο βιβλίο. Εδώ αναφέρεται μια ελάχιστη (και σωστά οργανωμένη) ομάδα

Τέλος, μπορούμε να αναφέρουμε και τους ρόλους στους οποίους εντάσσονται οι ενδιαφερόμενοι για το έργο (stakeholders), αν και δεν αποτελούν μέλη της ομάδας έργου. Αυτοί είναι:

Συνοπτικά, θα τους λέγαμε «ομάδα έργου του πελάτη»

- Υποστηρικτής έργου (project sponsor)
  Ένα ή περισσότερα άτομα που αρχίζουν και εγκρίνουν το έργο και το αποτέλεσμά του
- Τελικός χρήστης
- Επιχειρησιακός υποστηρικτής (customer ή business sponsor)
  Ένα ή περισσότερα άτομα που περιμένουν να κερδίσουν επιχειρησιακή αξία από το έργο
- Υπεύθυνοι λειτουργίας
  Αυτοί που είναι υπεύθυνοι για την λειτουργία της λύσης, αφού παραδοθεί

Πίνακας προσαρμογής (tradeoff matrix)

Σε ένα έργο, εξετάζουμε τρεις παράγοντες:

- Την - επιθυμητή - λειτουργικότητα (features)
- Τους - διαθέσιμους - πόρους (resources)
- Το χρονοδιάγραμμα (schedule)

Προσαρμόζουμε το έργο, θεωρώντας τον ένα παράγοντα σταθερό (μη διαπραγματεύσιμο), και ρυθμίζουμε (επιλέγουμε λειτουργικότητα, πόρους ή χρονοδιάγραμμα) ένα από τους δύο άλλους. Τότε ο τρίτος μεταβάλλεται όπως απαιτείται.
Για παράδειγμα: Με δεδομένους πόρους, θα επιλέξουμε την λειτουργικότητα και θα μεταβάλλουμε το χρονοδιάγραμμα.

Επαναλήψεις και εκδόσεις (iteration & versioned releases)

Εξελίσσουμε ένα έργο παράγοντας εκδόσεις/υποέργα. Για κάθε έκδοση ακολουθούμε όλα τα στάδια της διαδικασίας.

Οραματισμός (envisioning phase)

Κατά το στάδιο αυτό, ουσιαστικά, «ορίζουμε» το έργο. Δηλαδή, οργανώνουμε την ομάδα έργου και

Συλλογή πληροφοριών, σε συνεργασία με τον πελάτη (την ομάδα του)

- Εκτιμούμε την παρούσα κατάσταση, για να ορίσουμε το προς επίλυση πρόβλημα
- Εντοπίζουμε τους επιχειρησιακούς στόχους του έργου (business goals)
- Διαπιστώνουμε τις κατηγορίες (user profiles) των χρηστών της λύσης
- Καταγράφουμε τις λειτουργικές απαιτήσεις (requirements)
- Καταγράφουμε τους κινδύνους (τα πιθανά προβλήματα) που επηρεάζουν την υλοποίηση (risk assessment)

με στόχο

- Να καθορίσουμε την διοικητική δομή (σχέδιο) του έργου (project structure)
- Να καθορίσουμε τον σκοπό και το εύρος του έργου (vision & scope)
- Να ορίσουμε το περίγραμμα της λύσης (solution concept)

Έγγραφα εργασίας

Οι πληροφορίες που συγκεντρώνονται καταχωρούνται σε έγγραφα εργασίας. Κάποια από αυτά μπορούν να αποτελέσουν συμπληρωματικά παραδοτέα (σημειώνονται με «ΠΠ»), ανάλογα με το έργο. Αυτά είναι:

- Αρχική κατάσταση από στοιχεία λειτουργικότητας προς δοκιμή - ΠΠ
- Αρχικό διάγραμμα use cases[3] - ΠΠ
- Αρχική κατάσταση απαιτήσεων – ΠΠ
- Γραφική απεικόνιση περιβάλλοντος χρήστη (GUI storyboard) – ΠΠ
- Κατάλογος τύπων χρηστών και εξωτερικών οντοτήτων (actors των use cases)
- Κατάλογος κανόνων (business rules)
- Ευρετήριο ορολογίας

Χρησιμοποιήστε ένα αρχείο MS Excel, με τα φύλλα (και τις στήλες):

1. Actors
    a. Ονομασία (με την οποία θα αναφέρεται στο εξής και θα χρησιμοποιηθεί στα use cases και στα σενάρια χρήσης [Τα σενάρια χρήσης είναι αναλυτικές περιγραφές των use cases και περιγράφονται στην ενότητα *Δημιουργία σεναρίων χρήσης (usage scenarios)* στην σελίδα 8]
    b. Περιγραφή
    c. Άλλες ονομασίες
    d. Σχετίζεται με (άλλους actors, ιεραρχίες κ.λπ.)
2. Απαιτήσεις
    a. Κωδικός (ID)
    b. Σύντομη περιγραφή (Τίτλος)
    c. Αναλυτική περιγραφή
    d. Κατηγορία απαίτησης (επιχειρησιακή, χρηστική, λειτουργική, ολοκλήρωσης κ.λπ. – εξαρτάται από το έργο) [Δείτε την ενότητα *Εξειδίκευση απαιτήσεων* στην σελίδα 8]
3. Κανόνες
    a. Κωδικός (ID)
    b. Σύντομη περιγραφή (Τίτλος)
    c. Αναλυτική περιγραφή
    d. Περιγραφή της διαδικασίας εφαρμογής
4. Ορολογία
    a. Όρος
    b. Σημασία

Προσθέστε άλλες στήλες, ανάλογα με τις ανάγκες και το μέγεθος του έργου. Κατά τις αρχική συλλογή πληροφοριών, η στήλη «Πηγή» (της πληροφορίας) βοηθά στον έλεγχο και την διασταύρωση …

Κομβικά σημεία

Τα ενδιάμεσα κομβικά σημεία είναι:

1. Ολοκλήρωση ομάδας έργου
2. Αρχική έκδοση του εγγράφου «Σκοπός & εύρος έργου»

και το στάδιο ολοκληρώνεται με την τελική έγκριση των παραδοτέων.

Παραδοτέα

Τα παραδοτέα του σταδίου είναι:

1. Το έγγραφο «Σχέδιο του έργου»
2. Το έγγραφο «Σκοπός & εύρος έργου»
3. Το έγγραφο «Ανάλυση κινδύνων»

Με βάση αυτά τα έγγραφα, προχωράμε στο επόμενο στάδιο.

Το έγγραφο «Σχέδιο του έργου»

Περιλαμβάνει:

- την ομάδα έργου, με ανάλυση των αρμοδιοτήτων κάθε μέλους (δηλαδή τους ρόλους στους οποίους εντάσσονται)
- την κατάσταση των φυσικών προσώπων που αποτελούν την ομάδα

[3] Είναι χρήσιμο να δίνονται κωδικοί (IDs) στα use cases, χρησιμοποιούνται στο επόμενο στάδιο

- το αρχικό χρονοδιάγραμμα[4]
- την αρχική εκτίμηση κόστους
- την κατάσταση των παραδοτέων[5]
- τον πίνακα προσαρμογής (tradeoff matrix)
- την μεθοδολογία ενσωμάτωσης αλλαγών (στην λειτουργικότητα, στο τι θα μείνει για επόμενη έκδοση κ.λπ.)
- τυχόν περιορισμούς (διαθεσιμότητας ατόμων και πόρων, ειδικές απαιτήσεις ολοκλήρωσης κ.λπ.)

Το έγγραφο «Σκοπός & εύρος έργου»

Περιλαμβάνει:

- Την περιγραφή του προβλήματος με αναφορά στην παρούσα κατάσταση (problem statement)
- Τον σκοπό του έργου (σε μία φράση, το vision statement)[6]
- Το εύρος του έργου, δηλαδή τι θα υλοποιηθεί σε αυτή την έκδοση/υποέργο και τι θα μείνει για επόμενες εκδόσεις/υποέργα (project scope)
- Τον στόχο (goal) του έργου, με αναφορά:
    - στο γενικό αρχιτεκτονικό περίγραμμα της λύσης (solution concept)
    - στις κατηγορίες χρηστών (user profiles)
    - στο συνοπτικό διάγραμμα χρήσης (high-level use cases)
    - στον αρχικό πίνακα απαιτήσεων (requirements), κατηγοριοποιημένο κατά τύπο

Το έγγραφο «Ανάλυση κινδύνων»

Στο έγγραφο αυτό καταγράφουμε και σχεδιάζουμε την αντιμετώπιση των πιθανών προβλημάτων (risks) που μπορεί να αντιμετωπίσουμε[7].

Χρησιμοποιείστε ένα φύλλο MS Excel, με τις στήλες:

1. Κωδικός (ID)
2. Περιγραφή κινδύνου
3. Συνέπεια κινδύνου
4. Πιθανότητα (possibility) – χρησιμοποιείστε μία κλίμακα, π.χ. 1-4 ή 1%-100%
5. Σοβαρότητα/επίδραση (severity/impact) – χρησιμοποιείστε μία κλίμακα π.χ. 1-4 ή 1-10 (& 100 για «καταστροφή»)
6. Βαθμός έκθεσης (exposure). Προκύπτει σαν το γινόμενο των δύο προηγουμένων. <u>Η ταξινόμηση ως προς αυτή την στήλη μας δείχνει ποια σημεία επικινδυνότητας πρέπει να αντιμετωπίσουμε πρώτα</u>
7. Σχέδια μετρίασης του κινδύνου (mitigation plan). Δηλαδή, τι πρέπει να γίνει για να μειωθεί η πιθανότητα του κινδύνου.
8. Σχέδια αντιμετώπισης των συνεπειών (contingency plan). Δηλαδή, τι πρέπει να γίνει εάν προκύψει ο κίνδυνος, για να μειωθούν οι συνέπειες.

Προσθέστε άλλες στήλες, ανάλογα με τις ανάγκες και το μέγεθος του έργου (π.χ. «Υπεύθυνος», «Κατηγορία» κ.λπ.)

Τα περιεχόμενα αυτού του εγγράφου «εξελίσσονται», καθώς η ανάλυση των κινδύνων συνεχίζεται και στο επόμενο στάδιο της διαδικασίας.

---

[4] Gantt chart, από MS Project

[5] Όλα τα παραδοτέα, και αυτά των επομένων σταδίων

[6] Τα χαρακτηριστικά ενός καλά δοσμένου σκοπού είναι SMART (specific, measurable, achievable, relevant, time-based), δηλαδή σαφής, μετρήσιμος, εφικτός, συναφής και χρονικά καθορισμένος

[7] Εδώ, περιλαμβάνονται και οι «απειλές» (threats), στις οποίες γίνεται αναφορά στην ενότητα *Ανάλυση απειλών (threat modeling)* στην σελίδα 12, αλλά σε πολύ γενική μορφή.

Ανάλυση (planning phase)

Το στάδιο της ανάλυσης είναι αυτό κατά το οποίο «σχεδιάζουμε» την λύση. Δηλαδή, ολοκληρώνουμε την συλλογή πληροφοριών και προχωράμε:

- στον εννοιολογικό σχεδιασμό (conceptual design)
- στον λογικό σχεδιασμό (logical design)
- στον φυσικό σχεδιασμό (physical design)

με στόχο να ετοιμάσουμε τις πλήρεις λειτουργικές προδιαγραφές της λύσης (functional specification).

Ο εννοιολογικός σχεδιασμός προηγείται του λογικού σχεδιασμού.
Ο λογικός σχεδιασμός προηγείται του φυσικού σχεδιασμού.
Μπορούμε να έχουμε παραλληλία, ανάλογα με το έργο και τα τμήματα/υποέργα του.

Έγγραφα εργασίας

Τα έγγραφα εργασίας από το προηγούμενο στάδιο εξελίσσονται (συμπληρώνονται και το περιεχόμενό τους αναλύεται σε μεγαλύτερο βαθμό). Παράλληλα, προστίθενται νέα έγγραφα (κυρίως διαγράμματα).

Από όλα αυτά τα έγγραφα που συγκεντρώνονται θα προκύψουν οι λειτουργικές προδιαγραφές.

Κομβικά σημεία

Τα ενδιάμεσα κομβικά σημεία είναι:

1. Ολοκλήρωση αποτίμησης τεχνολογιών (technology validation complete)
   Γίνεται εκτίμηση και επιλογή των τεχνολογιών (και ετοίμων προϊόντων) που θα χρησιμοποιηθούν στην σχεδιαζόμενη λύση και έλεγχος του διαθεσίμου υλικού και λογισμικού του πελάτη (τοπολογία δικτύου, διαθέσιμο υλικό κ.λπ.)
2. Ολοκλήρωση λειτουργικών προδιαγραφών
3. Ολοκλήρωση πλάνου & χρονοδιαγράμματος του έργου
4. Ολοκλήρωση υποδομής ανάπτυξης και δοκιμών

και το στάδιο ολοκληρώνεται με την τελική έγκριση των παραδοτέων.

Παραδοτέα

Τα παραδοτέα του σταδίου είναι:

1. Το έγγραφο «Σύνοψη λειτουργικών προδιαγραφών» (functional specification summary)
2. Το έγγραφο «Πλάνο και χρονοδιάγραμμα του έργου» (master project plan & master project schedule)
3. Το έγγραφο «Σχέδιο ασφαλείας» (security plan)

Εννοιολογικός σχεδιασμός

Ορίζουμε το πρόβλημα με use cases και σενάρια χρήσης

Κατά τον εννοιολογικό σχεδιασμό, αντιμετωπίζουμε το πρόβλημα από την σκοπιά του πελάτη. Επεξεργαζόμαστε τις πληροφορίες που συγκεντρώσαμε κατά το προηγούμενο στάδιο, ξεκαθαρίζουμε ασαφή σημεία, και τις αναλύουμε και εξειδικεύουμε.

Τα ασαφή σημεία λειτουργικότητας και τα τυχόν εξωγενή προβλήματα (π.χ. διαθεσιμότητα απαραιτήτων ατόμων) που εντοπίσαμε και καταγράψαμε στο έγγραφο «Ανάλυση κινδύνων» πρέπει να αντιμετωπισθούν στην αρχή της φάσης αυτής. Οι απειλές (threats) αντιμετωπίζονται κατά τον φυσικό σχεδιασμό.

Κατά την διάρκεια του εννοιολογικού σχεδιασμού, είμαστε σε επαφή με τον πελάτη, και, πριν προχωρήσουμε στον λογικό σχεδιασμό, πιστοποιούμε, σε συνεργασία μαζί

του, ότι έχουμε επιλύσει σωστά το πρόβλημα. Δηλαδή, ότι οι απαιτήσεις, τα use cases και τα σενάρια χρήσης απεικονίζουν ορθά τις απαιτήσεις της λύσης.
Οι επόμενες ενότητες περιγράφουν τις διαδικασίες στις οποίες προχωράμε.

Επιλογή αρχιτεκτονικής

Η επιλογή αρχιτεκτονικής εξαρτάται από την αποτίμηση τεχνολογιών και ετοίμων προϊόντων, από συγκεκριμένες απαιτήσεις του πελάτη και από τις διαφαινόμενες απαιτήσεις της λύσης.

Τα τυπικά μοντέλα είναι:

- client / server
- n-tier
- stateless [server] – statefull [client]

και, μην παραλείψετε να σκεφτείτε την δυνατότητα caching[8].

Πρακτικά:

- δείτε τι client (fat, thin ή smart) «ταιριάζει», ανάλογα με τον τύπο της λύσης και τις ανάγκες τήρησης κατάστασης (state) σε αυτόν.
- δείτε τις ανάγκες πρόσβασης δεδομένων
- δείτε τις «υποχρεωτικές» κατευθύνσεις
- δείτε τις ανάγκες ολοκλήρωσης με άλλα συστήματα

και παρεμβάλετε όσα στρώματα (tiers) χρειάζεται ανάμεσα σε client και data store.

Είναι σημαντικό κάθε στρώμα να «μιλάει» μόνο με τα γειτονικά του.

Εξειδίκευση απαιτήσεων

Αναλύουμε και εξειδικεύουμε τον πίνακα απαιτήσεων (που έχουμε έτοιμο από το προηγούμενο στάδιο), έτσι ώστε να καταλήξουμε στον πλήρη πίνακα απαιτήσεων. Αυτός πρέπει να ελεγχθεί σε συνεργασία με τον πελάτη.

Οι απαιτήσεις στις οποίες καταλήγουμε πρέπει:

- να είναι καλά ορισμένες
- να είναι απλές (να αντιμετωπίζουν ένα και μόνο ζήτημα)
- να είναι ελέγξιμες
- να είναι οργανωμένες ιεραρχικά (σε ομάδες συναφών απαιτήσεων οι οποίες βρίσκονται κάτω από μία γενικότερη)
- να εκφράζονται με την ορολογία του προβλήματος

Είναι δυνατή η χρήση του φύλλου MS Excel που αναφέρθηκε στο προηγούμενο στάδιο, αναλύοντας τις γενικές απαιτήσεις σε ειδικότερες.
Επίσης, μπορούν να χρησιμοποιηθούν εργαλεία brainstorming.

Είναι σημαντικό να κατηγοριοποιηθούν οι απαιτήσεις σε:

- χρηστικές (user requirements)
  Είναι αυτές που δεν έχουν σχέση με την λειτουργικότητα της λύσης αλλά ορίζουν τον τρόπο αλληλεπίδρασης των χρηστών με το σύστημα. Αποτελούν οδηγό για την διαμόρφωση του περιβάλλοντος χρήστη
- συστήματος (system requirements)
  Αναφέρονται σε αλληλεπίδραση με εξωτερικά συστήματα
- λειτουργίας (operations requirements)

[8] Στο βιβλίο (που αποτέλεσε το έναυσμα για την συγγραφή του παρόντος), αναφέρεται και η “Layered-client-cache-stateless-cache-server”. Δηλαδή η browser-based παραλλαγή της Windows DNA αρχιτεκτονικής.

Οι σχετικές με ασφάλεια, διαθεσιμότητα και δυνατότητες διαχείρισης, κλιμάκωσης και υποστήριξης
- επιχειρησιακές (business requirements)
  Οι απαιτήσεις λειτουργικότητας (το τι πρέπει να κάνει η λύση)

Εξειδίκευση use cases

Αναλύουμε και συμπληρώνουμε τα use cases του προηγουμένου σταδίου, δημιουργώντας θυγατρικές (subordinate) use cases.

Πρακτικά, σπάμε κάθε use case στα τμήματά του (σε ομάδες βημάτων που έχουν έννοια και μόνες τους) , και αυτά τα κάνουμε θυγατρικά use cases. Φροντίζουμε να εντοπίσουμε σωστά τους actors και τις συσχετίσεις.

Παράγουμε τα σχετικά διαγράμματα με χρήση ενός εργαλείου UML modeling[9].

Δημιουργία σεναρίων χρήσης (usage scenarios)

Αυτά δημιουργούνται για τα θυγατρικά use cases που ετοιμάστηκαν κατά την εξειδίκευση.

Είναι δομημένες περιγραφές των απλών βημάτων που υλοποιούν ένα θυγατρικό use case, σε μορφή κειμένου. Περιλαμβάνουν:
- Ονομασία use case
- Κωδικό (ID)[10]
- Κωδικό απαίτησης – Από τον πίνακα απαιτήσεων, συνήθως επιχειρησιακή ή συστήματος
- Σκοπός – γιατί υπάρχει;
- Περιγραφή – μια σύντομη περιγραφή
- Actors
- Σχετικά use cases – άλλα use cases με τα οποία συνδέεται[11]
- Προϋποθέσεις (preconditions) – τι οδήγησε στην ανάγκη εκτέλεσης του use case
- Κύρια πορεία (basic course) – τα στοιχειώδη βήματα, αριθμημένα
- Εναλλακτική πορεία (alternative course) – τυχόν «διακλαδώσεις» <u>μέσα στην ροή των βημάτων</u>, εάν υπάρχουν.
- Τελική κατάσταση (postconditions) – το τελικό αποτέλεσμα (και που οδηγείται ο χρήστης)
- Εκκρεμότητες (unresolved issues) – τυχόν σημεία προς διευκρίνηση

Έλεγχος και βελτιστοποίηση του εννοιολογικού σχεδιασμού

Όπως αναφέρθηκε και νωρίτερα, τα παραγόμενα του εννοιολογικού σχεδιασμού πρέπει να ελεγχθούν ώστε να πιστοποιηθεί ότι αντιμετωπίζουν σωστά το πρόβλημα.

Παράλληλα, πρέπει να έχουμε σαν στόχο την βελτιστοποίηση του σχεδιασμού. Αυτό σημαίνει ότι:
- πρέπει να λάβουμε υπ' όψη μας πιθανές μελλοντικές εκδόσεις της λύσης

[9] Το MS Visio υπάρχει στο VS.Net 2003 Enterprise Architect. Υπάρχουν και άλλα, ανεξάρτητα ή με δυνατότητα ενσωμάτωσης στο VS.Net 2003.
[10] Εάν είχαν δοθεί κωδικοί πρώτου επιπέδου στο προηγούμενο στάδιο, τώρα δίνονται δευτέρου επιπέδου (π.χ. εάν είχατε ID=UC2 στο στάδιο του οραματισμού, και σπάσατε το use case σε δύο θυγατρικά, έχετε τους κωδικός UC2.1 & UC2.2)
[11] Τα use cases με τα οποία υπάρχει σχέση "include/uses", "extends" κ.λπ.

- πρέπει να εξετάσουμε τα περιθώρια βελτίωσης των διαδικασιών που θα υλοποιεί το σύστημα[12]
- πρέπει να εντοπίσουμε τρόπους μείωσης του φόρτου του χρήστη

Οι παραπάνω ενέργειες γίνονται σε συνεργασία με τον πελάτη ο οποίος πρέπει να συμφωνήσει με τα προτεινόμενα.

Λογικός σχεδιασμός

Ορίζουμε το πρόβλημα σαν λογικό σύνολο αντικειμένων και υπηρεσιών

Κατά τον λογικό σχεδιασμό, αντιμετωπίζουμε το πρόβλημα από την σκοπιά της ομάδας έργου.

Προχωράμε στον εντοπισμό:
- αντικειμένων (business objects)
- υπηρεσιών (services)
- ιδιοτήτων (attributes)
- σχέσεων (relationships)

και παράγουμε το λογικό μοντέλο αντικειμένων (logical object model).

Παράλληλα, δημιουργούμε:
- το λογικό μοντέλο των δεδομένων
  Είναι ένα προσχέδιο της μορφής της βάσης δεδομένων, εάν βέβαια υπάρχει. Αυτό το βήμα δεν είναι υποχρεωτικό, ειδικά εάν το λογικό μοντέλο αντικειμένων δίνει την απαραίτητη πληροφορία.
  Εάν τα δεδομένα αποθηκεύονται με άλλο μηχανισμό, περιγράφουμε αυτό τον μηχανισμό (για παράδειγμα, XML Schemas και XML αρχείο με δείγμα δεδομένων
- το προσχέδιο του περιβάλλοντος χρήστη[13].

Τέλος επανεξετάζουμε (και, εάν απαιτείται, τροποποιούμε) την αρχιτεκτονική της λύσης.

Λογικό μοντέλο αντικειμένων (logical object model)

Το λογικό μοντέλο αντικειμένων αποτελεί το κύριο παράγωγο του λογικού σχεδιασμού, σε μορφή UML class diagram.

Με βάση τα σενάρια χρήσης, έχουμε:
- άνθρωποι ή πράγματα → αντικείμενα
- ενέργειες → υπηρεσίες των αντικειμένων
- χαρακτηριστικά των αντικειμένων → ιδιότητες
- συσχετίσεις μεταξύ των αντικειμένων → σχέσεις

Για να κάνουμε σαφέστερη την αλληλεπίδραση μεταξύ των αντικειμένων, χρησιμοποιούμε UML interaction diagrams (sequence & collaboration diagrams[14]).

Χρήσιμα είναι και τα activity diagrams, που μπορούν να περιγράψουν μια ροή εργασιών σε υψηλό επίπεδο.

---

[12] Ανάλογα με το έργο, μπορεί να απαιτηθεί BPR (Business Process Re-engineering)
[13] Κάποιες απεικονίσεις του περιβάλλοντος χρήστη (μπορούν να χρησιμοποιηθούν το MS PowerPoint ή/και το MS Visio). Αυτό το βήμα δεν είναι υποχρεωτικό, γίνεται εάν έχει έννοια (και έχουμε αρκετές πληροφορίες).
[14] Γενικά, αυτά τα διαγράμματα θεωρούνται ισόμορφα. Το sequence diagram τονίζει την χρονική σειρά και το collaboration diagram τις δομικές συσχετίσεις

Έλεγχος του λογικού σχεδιασμού

Ελέγχουμε τον λογικό σχεδιασμό, πιστοποιώντας:

- ότι καλύπτονται <u>όλες</u> οι απαιτήσεις
- ότι τα αντικείμενα είναι πλήρη από άποψη ιδιοτήτων και παρέχουν όλες τις απαιτούμενες υπηρεσίες
- ότι τα σενάρια χρήσης μπορούν να υλοποιηθούν

Φυσικός σχεδιασμός

Ορίζουμε πλήρως την λύση

Κατά τον φυσικό σχεδιασμό, <u>αντιμετωπίζουμε το πρόβλημα από την σκοπιά της ομάδας ανάπτυξης</u>.

Εξελίσσουμε την σχεδίαση έτσι ώστε να δώσουμε στην ομάδα ανάπτυξης πλήρης προδιαγραφές. Παράλληλα, το παραγόμενο υλικό θα συμπεριληφθεί (συνήθως περιληπτικά) στα παραδοτέα έγγραφα.

UML modeling

Παράγουμε αναλυτικά διαγράμματα που απεικονίζουν την σχεδιαζόμενη λύση. Αυτά θα χρησιμοποιηθούν από την ομάδα ανάπτυξης σαν βάση για τον κώδικα[15]. Δηλαδή:

- class & object diagrams
- component diagrams
- sequence, collaboration, activity & state diagrams
- deployment diagrams (δηλαδή network topology & component/data topology diagrams)

Περιβάλλον χρήστη (presentation layer)

Καταλήγουμε στην τελική μορφή του περιβάλλοντος χρήστη (windows-based, web-based, mobile-device-based ή document-based – ή και σε κάποιο συνδυασμό, ανάλογα με το έργο).

Παράγουμε (ή βελτιώνουμε και εμπλουτίζουμε, εάν ξεκινήσαμε την διαδικασία κατά τον λογικό σχεδιασμό) σχέδια του περιβάλλοντος χρήστη.

Μπορούμε να διαχωρίσουμε το περιβάλλον χρήστη σε τμήματα[16]:

- τα στοιχεία περιβάλλοντος χρήστη (user interface components)
  Διαχειρίζονται την αλληλεπίδραση με τον χρήστη
- τα στοιχεία διαδικασίας χρήστη (user process components)
  Διαχειρίζονται την όλη διαδικασία, δηλαδή τηρούν κατάσταση και δεδομένα. Επίσης, «οδηγούν» το περιβάλλον χρήστη.

Διαχείριση δεδομένων (data layer)

Καταλήγουμε στην τελική μορφή της αποθήκευσης δεδομένων, συνήθως σε κάποιο RDBMS και παράγουμε ERD διαγράμματα για την βάση δεδομένων.

Εξετάζουμε επίσης:

Μόνο εάν είναι σταθεροί

- την δυνατότητα χρήσης μηχανισμών του RDBMS για υλοποίηση επιχειρησιακών κανόνων[17]

[15] Υπάρχουν διάφορα εργαλεία που επιτρέπουν την παραγωγή σκελετού κώδικα από τα διαγράμματα

[16] Περαιτέρω επεξηγήσεις ξεφεύγουν από τον σκοπό του παρόντος κειμένου. Μελετήστε το Model-View-Controller pattern και τις παραλλαγές του

[17] Εδώ <u>δεν</u> εννοείται η ακεραιότητα δεδομένων (data integrity). Αυτή <u>πρέπει</u> να τηρείται στο RDBMS

- εάν απαιτείται η δημιουργία ενδιάμεσων στοιχείων (components) που να απομονώνουν το RDBMS από την υπόλοιπη λύση[18]

Προδιαγραφές ασφαλείας (security specifications)

Για να καταλήξουμε σε μια στρατηγική ασφάλειας[19], προχωράμε στα εξής βήματα:

1. Εντοπίζουμε τους πόρους που πρέπει να είναι ασφαλείς
2. Επιλέγουμε μια πολιτική εκχώρησης δικαιωμάτων (authorization)
3. Επιλέγουμε τις ταυτότητες (identities) που χρησιμοποιούνται για πρόσβαση στους πόρους
4. Εξετάζουμε εάν η ταυτότητα πρέπει να διατρέχει το σύστημα
5. Επιλέγουμε μια πολιτική ταυτοποίησης (authentication)

Λαμβάνουμε βέβαια υπ' όψη μας τις απαιτήσεις της λύσης.

Τέλος, σχεδιάζουμε τον μηχανισμό auditing, εάν απαιτείται.

Ανάλυση απειλών (threat modeling)[20]

Για να αντιμετωπίσουμε την κακόβουλη χρήση της λύσης που αναπτύσσουμε, πρέπει να προχωρήσουμε σε ανάλυση των τρόπων που είναι δυνατή η «επίθεση» κατά της λύσης.

Το τυπικό μοντέλο κατηγοριοποίησης είναι το STRIDE, όπου:

- **S** (Spoofing identity) – Ο κακόβουλος χρήστης παρουσιάζεται σαν νόμιμος χρήστης
- **T** (Tampering) – Αλλοίωση στοιχείων (δεδομένων, ρυθμίσεων κ.λπ.)
- **R** (Repudiation) – Αδυναμία απόδειξης ότι συγκεκριμένη ενέργεια έγινε από συγκεκριμένο χρήστη
- **I** (Information disclosure) – Ο κακόβουλος χρήστης μπορεί να δει δεδομένα που δεν πρέπει
- **D** (Denial of service) – Η λύση δεν μπορεί να εξυπηρετήσει το σκοπό της, λόγω ελλείψεως πόρων (που δημιουργεί ο κακόβουλος χρήστης)
- **E** (Elevation of privilege) – Ο κακόβουλος χρήστης αποκτά περισσότερα δικαιώματα από ότι προβλέπεται

Οι περισσότερες απειλές ανήκουν σε πολλαπλές κατηγορίες.

Για να εντοπίσουμε σωστά τις πιθανές απειλές πρέπει:

1. Να χωρίσουμε (decompose) την λύση στα συστατικά της τμήματα
2. Να εντοπίσουμε τις απειλές
3. Να κατηγοριοποιήσουμε και να βαθμολογήσουμε (ως προς την σοβαρότητα) τις απειλές
4. Να βρούμε τον τρόπο μείωσης τους.

Σε αυτά βοηθούν:

- Η μελέτη της ροής δεδομένων (με Data Flow Diagrams – DFDs)
- Η δημιουργία δέντρων απειλής (threat trees)
- Η κατηγοριοποίηση κατά STRIDE
- Η βαθμολόγηση κατά DREAD (**D**amage potential, **R**eproducibility, **E**xploitability, **A**ffected users, **D**iscoverability)

---

[18] Συνήθως απαιτείται εάν πρέπει να έχουμε την δυνατότητα αλλαγής RDBMS

[19] Εδώ, εννοείται η ασφάλεια της λύσης για «κανονική» χρήση, όχι για κακόβουλη.

[20] Για περισσότερες πληροφορίες και τεχνικές, δείτε τα βιβλία Writing secure code (ISBN 0-7356-1722-8) των M. Howard και D. LeBlanc) και Threat Modeling (ISBN 0-7356-1991-3) των F. Swiderski και W. Snyder

Τέλος, καταλήγουμε στις τεχνικές αντιμετώπισης τους.

Άλλα σημεία προσοχής

Παράλληλα με τα προηγούμενα, πρέπει να ληφθούν υπ' όψη:

- οι απαιτήσεις κλιμάκωσης (scalability)
- οι απαιτήσεις διαθεσιμότητας (availability)
- οι απαιτήσεις αξιοπιστίας (reliability)
- οι απαιτήσεις απόδοσης (performance)
- οι απαιτήσεις ολοκλήρωσης (interoperability)
- η ανάγκη υποστήριξης πολλαπλών γλωσσών (globalization & localization)
- Οι διαχειριστικές απαιτήσεις, δηλαδή
    - παρακολούθηση (monitoring)
    - μεταφορά δεδομένων από άλλα συστήματα (data migration)
    - προδιαγραφές αδειών χρήσης (licensing specifications)

Προετοιμασία των επομένων σταδίων

Καθώς ολοκληρώνεται το στάδιο της ανάλυσης, γίνεται η προετοιμασία για τα στάδια της ανάπτυξης, της σταθεροποίησης και της εγκατάστασης. Επίσης, οργανώνεται το υλικό που παρήχθη σε ένα έγγραφο, τις «Τεχνικές προδιαγραφές».

Προετοιμασία για το στάδιο της ανάπτυξης

Καθορίζουμε ότι είναι σχετικό με την εύρυθμη πορεία της ανάπτυξης. Τα κυριότερα σημεία είναι:

- γενική στρατηγική παράδοσης (overall design strategy) – τι παραδίδουμε πρώτα, εάν προχωράμε κατά πλάτος ή κατά βάθος κ.λπ.
- διαίρεση της εφαρμογής σε στοιχεία (components)
- μεθοδολογίες και τυποποιήσεις (guidelines & standards)
- διαχείριση εκδόσεων και πηγαίου κώδικα (versioning & source control)
- εργαλεία που θα χρησιμοποιηθούν
- εκπαίδευση της ομάδας ανάπτυξης
- design patterns

Παράλληλα, κατανέμουμε την εργασία και ετοιμάζουμε το πλάνο και το χρονοδιάγραμμα της ομάδας ανάπτυξης.

Προετοιμασία για το στάδιο της σταθεροποίησης[21]

Προετοιμάζουμε:

- το σχέδιο δοκιμών (test plan)
- το σχέδιο πιλοτικής λειτουργίας (pilot plan)

Το σχέδιο δοκιμών ελέγχει:

- τα στοιχεία της εφαρμογής (component/unit testing)
- την βάση δεδομένων (database testing)
- την ασφάλεια (security testing)
- την χρηστικότητα (usability testing)

και περιλαμβάνει:

- μεθοδολογία δοκιμών
- τι δοκιμάζεται

[21] Τα στοιχεία που αναφέρονται είναι τα, κατά την γνώμη μου, σημαντικότερα. Ανάλογα με το έργο, μπορεί να απαιτηθούν και άλλα (π.χ. stress testing σε ένα εμπορικό web site)

- τι αποτελέσματα αναμένονται
- πως γίνεται η αναφορά λαθών

Το σχέδιο πιλοτικής λειτουργίας περιγράφει τον τρόπο που η υποψήφια προς παράδοση (release candidate) έκδοση θα ελεγχθεί.

Προετοιμασία για το στάδιο της εγκατάστασης

Σχεδιάζουμε την μέθοδο εγκατάστασης της λύσης, καταγράφουμε τις απαραίτητες υποδομές, την μέθοδο εγκατάστασης, τις ανάγκες υποστήριξης και γενικά, ότι έχει σχέση με την επιτυχή εγκατάσταση της λύσης.
Τα ακριβή τμήματα του σχεδίου εγκατάστασης εξαρτώνται από το έργο.

Τεχνικές προδιαγραφές

Οι τεχνικές προδιαγραφές περιλαμβάνουν το σύνολο του υλικού που παρήχθη κατά το στάδιο της ανάλυσης, με κυριότερα:

- σύνοψη της αρχιτεκτονικής
- μοντέλο αντικειμένων (object model)[22]
- μοντέλα ροής (code flow) – περιγραφή του τι πρέπει να κάνει κάθε μέθοδος

Επίσης, περιλαμβάνουν:

- κωδικούς σφαλμάτων (error codes)
- μέθοδο καταγραφής σφαλμάτων (error logging)
- ρυθμίσεις (configuration) – εγγραφές στο μητρώο (registry) και ότι άλλο σχετικό

Το έγγραφο «Σύνοψη λειτουργικών προδιαγραφών»

Περιλαμβάνει:

- αναφορά στο σκοπό και το εύρος του έργου (συνοπτικά, δεν επαναλαμβάνουμε το έγγραφο «Σκοπός & εύρος του έργου»)
- ιστορικό του έργου, με αναφορά σε γεγονότα και αποφάσεις που οδήγησαν το έργο στην μορφή που έχει πλέον. Για παράδειγμα, αποφάσεις που άλλαξαν τις απαιτήσεις ή μετέβαλλαν τον διαθέσιμο χρόνο κ.λπ.
- σύνοψη του πίνακα απαιτήσεων (της τελικής του μορφής)
- σύνοψη του σχεδίου ασφαλείας[23]
- σύνοψη των μεθόδων ολοκλήρωσης με άλλα συστήματα
- σύνοψη των διαδικασιών εγκατάστασης και απεγκατάστασης

και ότι άλλο συμφωνηθεί να συμπεριλαμβάνει …[24]

Το έγγραφο «Πλάνο και χρονοδιάγραμμα του έργου»

Περιλαμβάνει την σύνθεση των πλάνων και χρονοδιαγραμμάτων των ομάδων εργασίας και το σχετικό συγκεντρωτικό χρονοδιάγραμμα[25].

Το έγγραφο «Σχέδιο ασφαλείας»

Περιλαμβάνει την σύνοψη:

---

[22] Ουσιαστικά, όλα τα UML διαγράμματα
[23] Ουσιαστικά, αναφορά στο σχετικό έγγραφο
[24] Συνήθως, η μορφή και το περιεχόμενο του εγγράφου αυτού είναι προκαθορισμένη από τον πελάτη
[25] Αυτό είναι το δεσμευτικό πλάνο & χρονοδιάγραμμα. Υπενθυμίζεται ότι το αρχικό χρονοδιάγραμμα (στο έγγραφο «Σχέδιο του έργου») έχει γίνει πριν το στάδιο της ανάλυσης, οπότε δεν μπορεί να προβλέψει με ακρίβεια ότι θα παρουσιασθεί

- της στρατηγικής ασφάλειας
- της στρατηγικής αντιμετώπισης απειλών

Δηλαδή, το περιγραφικό τμήμα και τις τελικές επιλογές που προέκυψαν από τις διαδικασίες δημιουργίας προδιαγραφών ασφαλείας και αντιμετώπισης απειλών.

Ανάπτυξη (developing phase)

Κατά το στάδιο αυτό, και αφού έχουν εγκριθεί τα παραδοτέα του προηγούμενου σταδίου (της ανάλυσης), υλοποιούμε την λύση.
Καθώς αναπτύσσεται ο κώδικας, κάνουμε και τις σχετικές δοκιμές. Ουσιαστικά, unit tests και integration tests. Αυτά θα γίνουν, σε ολοκληρωμένη μορφή στο επόμενο στάδιο.

Ο κώδικας που παράγεται σε αυτό το στάδιο ονομάζεται **alpha code**.

Σταθεροποίηση (stabilizing phase)

Κατά το στάδιο αυτό η προσθήκη λειτουργικότητας στην λύση έχει ολοκληρωθεί.

Ο κώδικας πλέον ονομάζεται **beta code**. Αλλαγές στον κώδικα γίνονται μόνο για διόρθωση λαθών[26].

Κομβικά σημεία

Τα ενδιάμεσα κομβικά σημεία είναι:

1. Σημείο σύγκλισης σφαλμάτων (bug convergence)
   Είναι το χρονικό σημείο όπου τα σφάλματα που διορθώνονται είναι περισσότερα από τα καινούργια που εμφανίζονται
2. Σημείο χωρίς σφάλματα (zero-bug bounce)
   Είναι το χρονικό σημείο που δεν υπάρχουν σφάλματα προς διόρθωση (ανεξάρτητα του εάν εμφανισθούν μετά σφάλματα)
3. Υποψήφιες (προς διανομή) εκδόσεις (release candidates)
   Αυτές παραδίδονται για δοκιμή ή/και πιλοτική λειτουργία
4. Χρυσή έκδοση (golden release)
   Μία από τις προς διανομή εκδόσεις, συνήθως η τελευταία. Είναι αυτή που θεωρείται το τελικό προϊόν

και το στάδιο ολοκληρώνεται με την τελική έγκριση των παραδοτέων.

Παραδοτέα

Πέραν του τελικού (προς διανομή ή εγκατάσταση) προϊόντος, είναι:

- τα συμπεράσματα της πιλοτικής λειτουργίας
- ο τελικός πηγαίος κώδικας[27]
- ο μηχανισμός (scripts, MSI packages) της εγκατάστασης
- τα εγχειρίδια χρήσης και το εκπαιδευτικό υλικό

Εγκατάσταση (deploying phase)

Αποτελεί το τελικό στάδιο της διαδικασίας. Περιλαμβάνει:

1. Προετοιμασία της θέσης εγκατάστασης
   Προγραμματισμός (χρονικός και υλικού) και πιστοποίηση ότι ο απαραίτητος εξοπλισμός είναι διαθέσιμος[28]

[26] Οι διαδικασίες κατηγοριοποίησης των λαθών, όπως και οι διαδικασίες testing ξεφεύγουν από τον σκοπό του παρόντος κειμένου
[27] Εάν προβλέπεται κάτι τέτοιο

2. Εγκατάσταση
   Η καθαυτό διαδικασία της εγκατάστασης και πιστοποίηση ότι όλα λειτουργούν όπως πρέπει. Ενεργοποίηση της λύσης για πρόσβαση από τους χρήστες
3. Εκπαίδευση
   Εκπαίδευση των τελικών χρηστών και του προσωπικού τεχνικής υποστήριξης της λύσης
4. Σταθεροποίηση
   Παρακολούθηση, για κάποιο χρονικό διάστημα, της ορθής λειτουργίας της λύσης

Συνήθως ακολουθεί και μία «ήσυχη» περίοδος (quiet period), όπου:

- γίνεται μεταφορά των αρμοδιοτήτων στην ομάδα υπευθύνων λειτουργίας
- συλλέγονται στατιστικά στοιχεία, για σχεδιασμό SLAs (Service Level Agreements)
- γίνονται οι τυπικές διαδικασίες ολοκλήρωσης του έργου

Τέλος, υπογράφεται η επίσημη αποδοχή της λύσης από τον πελάτη.

Πρακτικές παρατηρήσεις

- Χρησιμοποιείστε όσα διαγράμματα είναι απαραίτητα για να περιγράψετε την λύση, μην κάνετε διαγράμματα για το προφανές…
- Πολλές φορές, μια καλογραμμένη περιγραφή είναι σαφέστερη από ένα διάγραμμα, ειδικά εάν οι developers δεν είναι καλοί «αναγνώστες» UML
- Σε κάποιο χρονικό σημείο, προς το τέλος του λογικού σχεδιασμού πρέπει να λάβετε υπόψη τον φυσικό διαχωρισμό της λύσης σε components (π.χ. ένα web service και ένα smart client). Ο φυσικός σχεδιασμός, και ειδικότερα η ανάλυση απειλών δεν μπορούν να γίνουν σε επίπεδο λύσης, γίνονται σε επίπεδο component
- Ο φυσικός διαχωρισμός συνήθως σημαίνει την εισαγωγή νέων αντικειμένων στο μοντέλο [αντικειμένων], με συνηθέστερα τα Remote Façade και Data Transfer Object

Επίλογος

Στο παρόν κείμενο, το κύριο βάρος δόθηκε στα δύο πρώτα στάδια της διαδικασίας (μεθοδολογίας) MSF.

Το λοιπά στάδια, όπως και η επιτυχία του έργου, εξαρτώνται απόλυτα από αυτά. Επίσης, τα τελικά παραδοτέα των τριών τελευταίων σταδίων εξαρτώνται, πολλές φορές, από τις απαιτήσεις του πελάτη (π.χ. εάν θα παραδοθεί ο πηγαίος κώδικας).

Έγινε προσπάθεια να μην παραλειφθούν σημαντικά σημεία της όλης διαδικασίας. Επίσης, η περιγραφή υπονοεί ότι ο σκοπός του έργου είναι η υλοποίηση μιας κατά παραγγελία (custom) λύσης. Όμως, η διαδικασία εφαρμόζεται και για παραγωγή ετοίμου προϊόντος, όπου ο πελάτης είναι η εταιρεία που το αναπτύσσει και θα το διαθέσει στην αγορά.

---

[28] Υπενθυμίζεται ότι ο αρχικός έλεγχος έχει γίνει κατά την φάση της ανάλυσης, καθώς και τυχόν απαραίτητες παραγγελίες υλικού και λογισμικού