



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Τεχνολογία Λογισμικού, 7ο/9ο εξάμηνο 2018-2019

Τεχνολογία Λογισμικού

Ν.Παπασπύρου, Αν.Καθ. ΣΗΜΜΥ, nickie@softlab.ntua.gr

Β.Βεσκούκης, Αν.Καθ. ΣΑΤΜ, v.vescoukis@cs.ntua.gr

Κ.Σαΐδης, ΠΔ 407, saiko@softlab.ntua.gr



How the customer explained it



How the project leader understood it



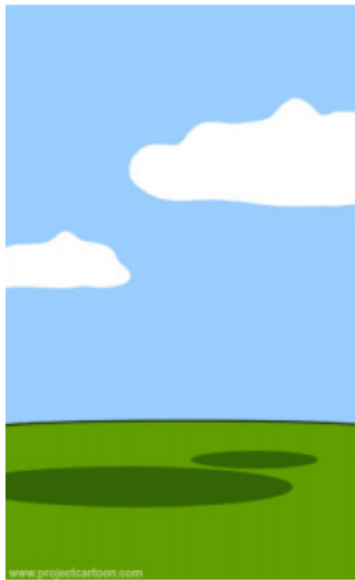
How the analyst designed it



How the programmer wrote it



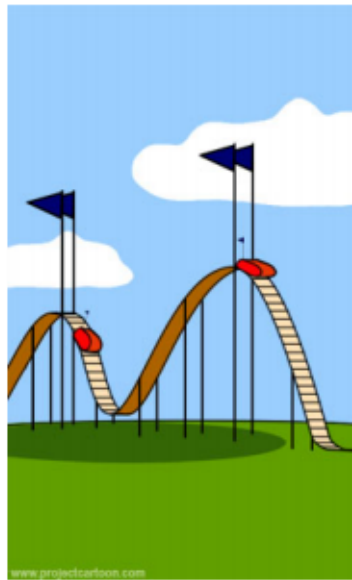
How the business consultant described it



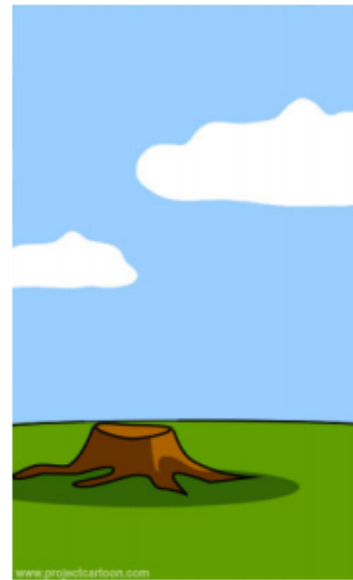
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

Κύκλος ζωής λογισμικού

Κύκλος ζωής Λογισμικού

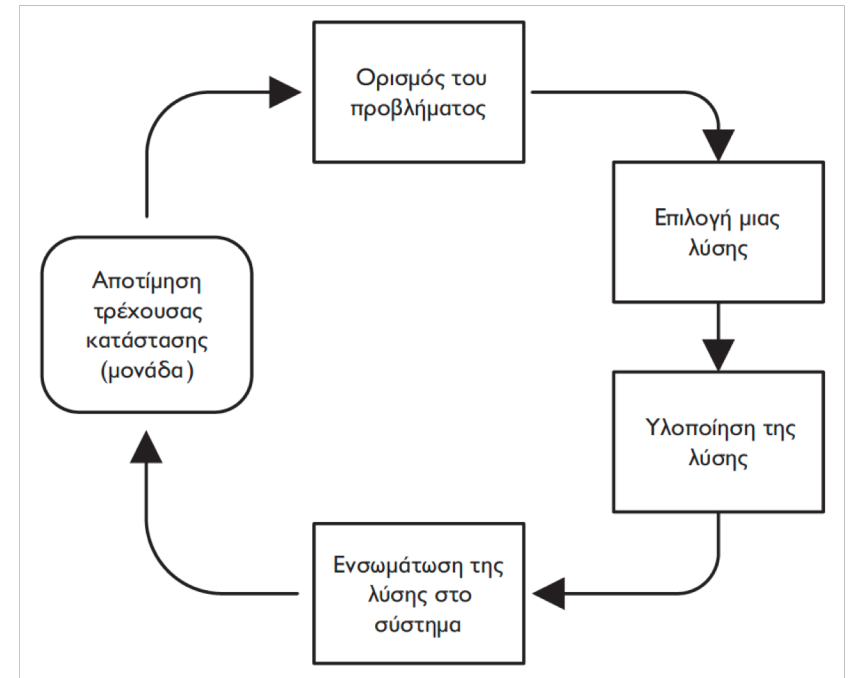
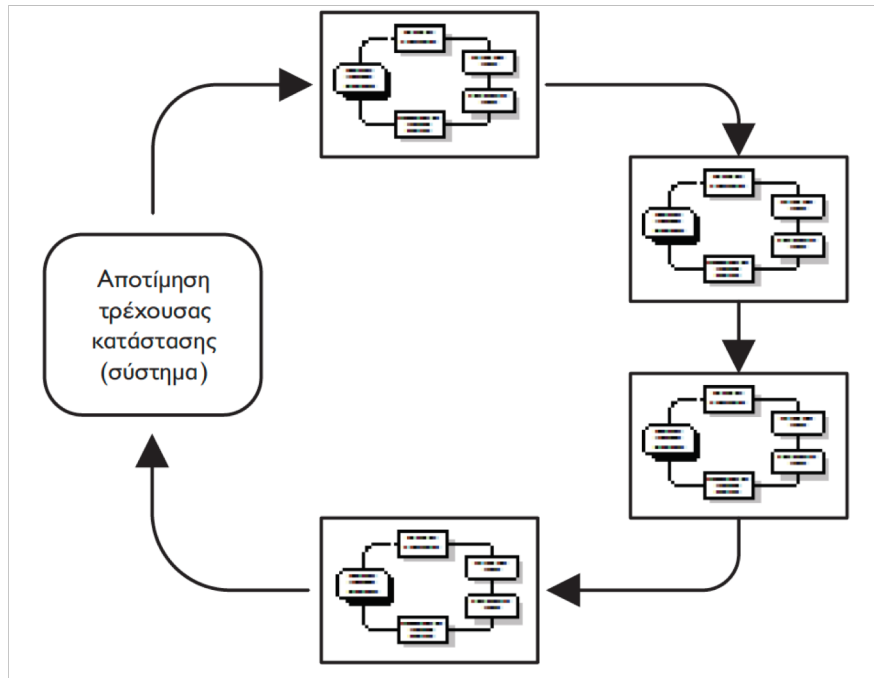
Οι φάσεις από τις οποίες διέρχεται μια εφαρμογή Λογισμικού

- από την σύλληψή της
- κατά την κατασκευή, χρήση, συντήρησή της
- μέχρι την απόσυρσή της

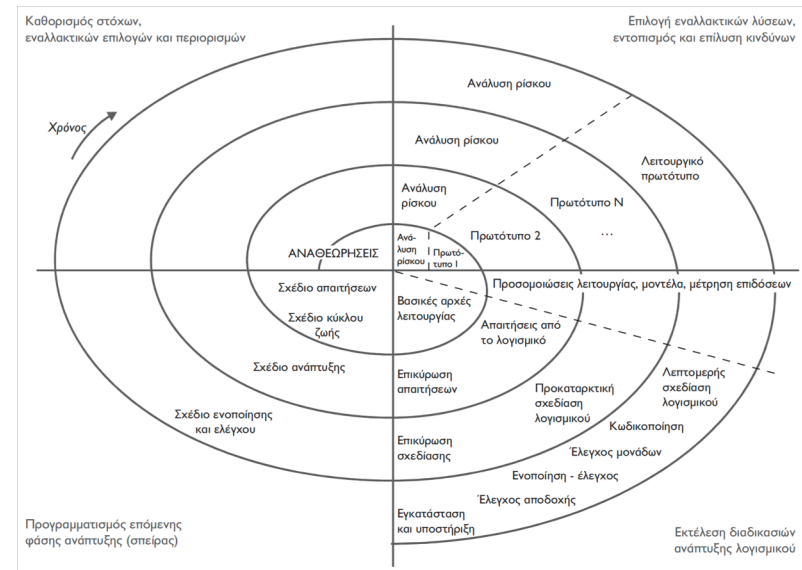
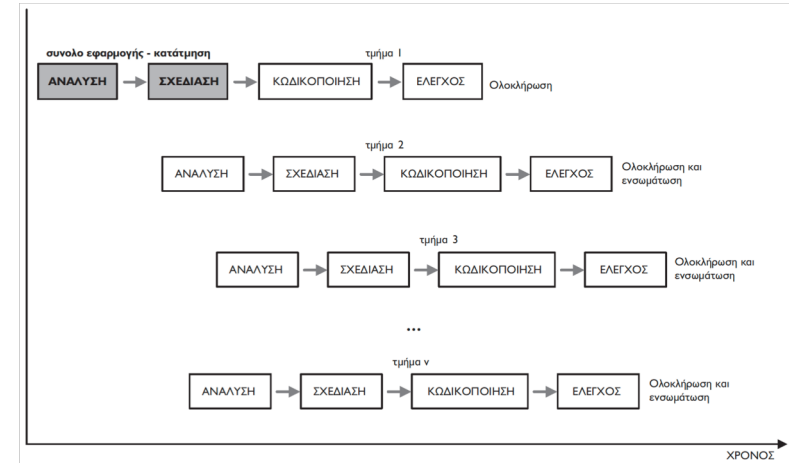
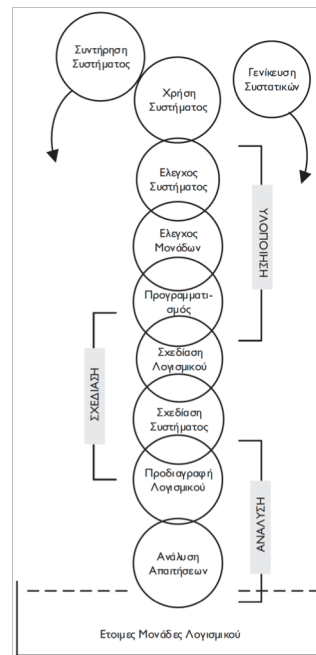
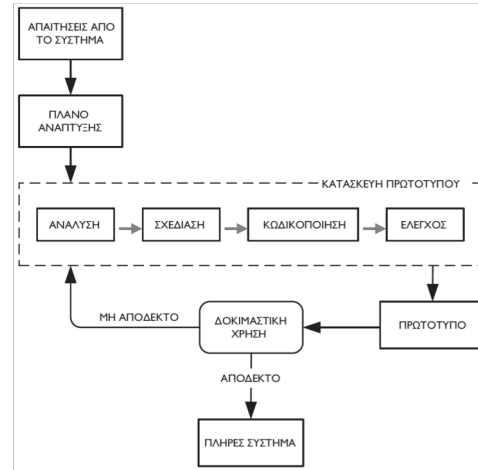
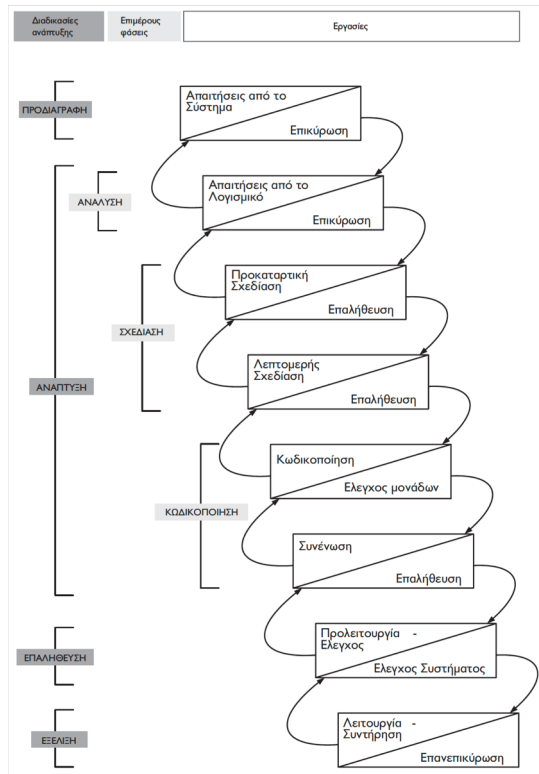
Ενα **μοντέλο κύκλου ζωής** περιγράφει

- Ποιες είναι αυτές οι φάσεις
- Ποιες εργασίες γίνονται και ποια προϊόντα παράγονται σε καθεμία
- Πώς διαδέχονται η μία την άλλη
- Σε ποια έκταση (του λογισμικού) εκτελούνται

Problem solving, iterative, creative!...



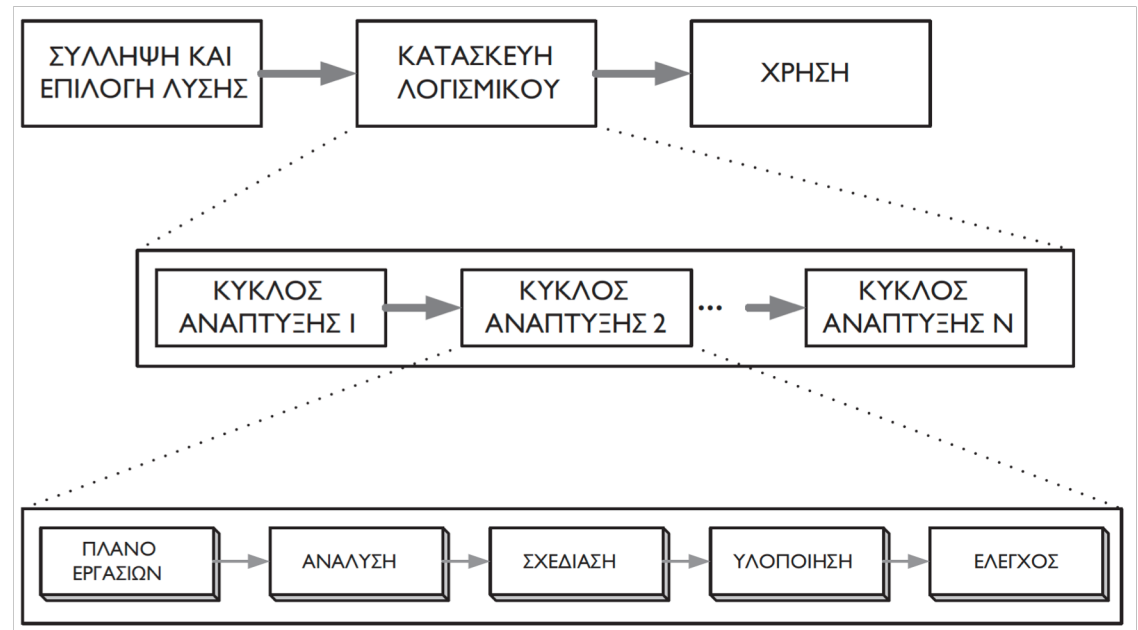
Κύκλος ζωής λογισμικού



Ένα γενικό μοντέλο κύκλου ζωής

Ανάπτυξη σε κύκλους σύμφωνα με τα χαρακτηριστικά και τις δυνατότητες του κατασκευαστή.

Γενικευμένη μορφή των προηγούμενων μοντέλων κύκλου ζωής.





Rational Unified Process (RUP)

Rational Unified Process

Γενικά

- Μια ολοκληρωμένη διαδικασία ανάπτυξης λογισμικού από την εταιρία Rational Corporation (αγοράστηκε από την IBM)
- Επαναληπτική (iterative) μεθοδολογία-πλαίσιο, δηλαδή απαιτεί εξειδίκευση
- Δυνατότητα εφαρμογής σε μικρά έως μεγάλα έργα λογισμικού
- Χρησιμοποιεί τη UML ως γλώσσα μοντελοποίησης

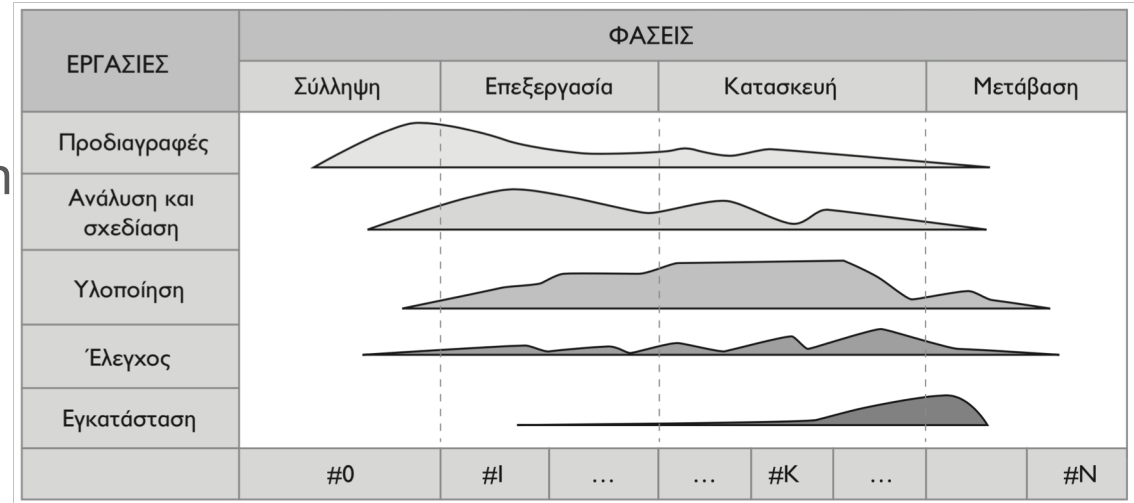
Έννοια-κλειδί: iterative

- Πολλές επαναληπτικές διαδικασίες που διαδέχονται η μία την άλλη
- Κάθε επανάληψη περιλαμβάνει ανάλυση απαιτήσεων, σχεδίαση και υλοποίηση
- Οι επαναλήψεις έχουν συγκεκριμένο χρονικό πλαίσιο

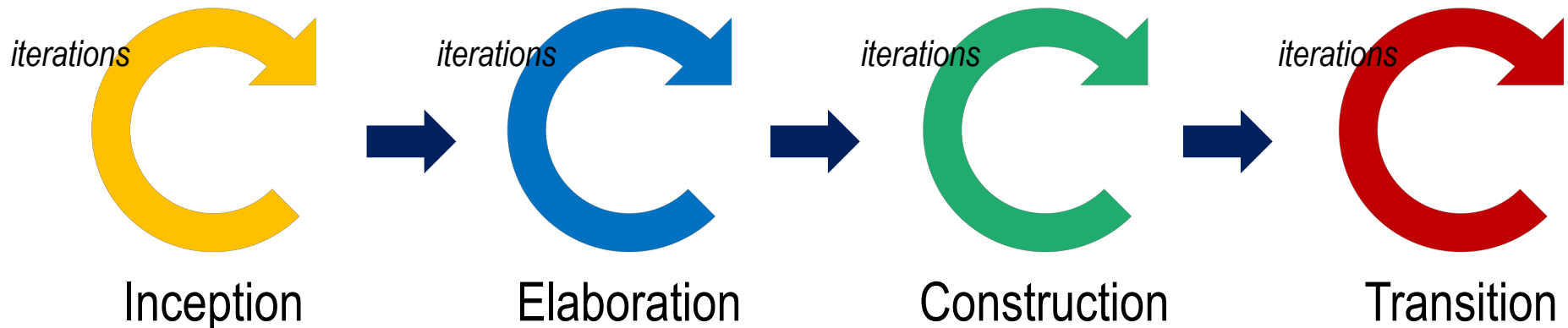
Φάσεις στη RUP

4 γενικές φάσεις

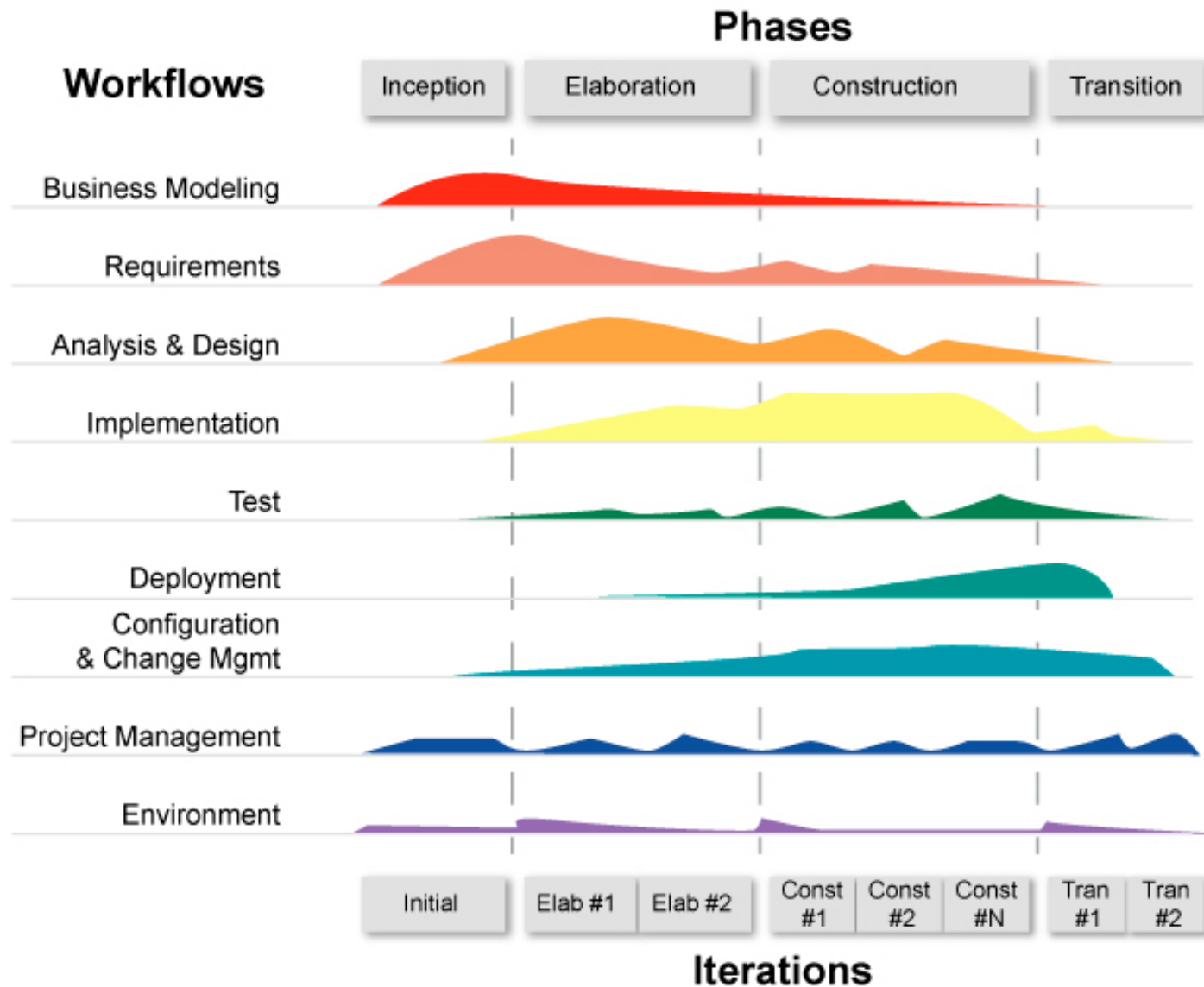
- Σύλληψη της ιδέας (inception)
- Επεξεργασία (elaboration)
- Κατασκευή (construction)
- Μετάβαση (transition)



Σε κάθε φάση (1 εβδομάδα έως μήνες) γίνονται επαναλήψεις (iterations).
Σε κάθε επανάληψη παράγεται ένα τμήμα λογισμικού

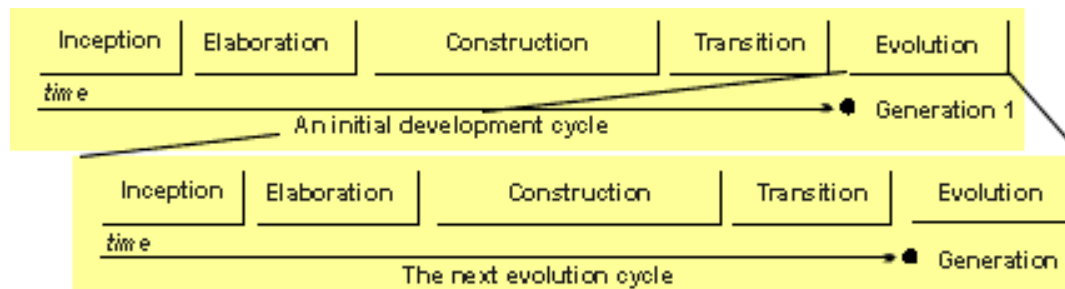
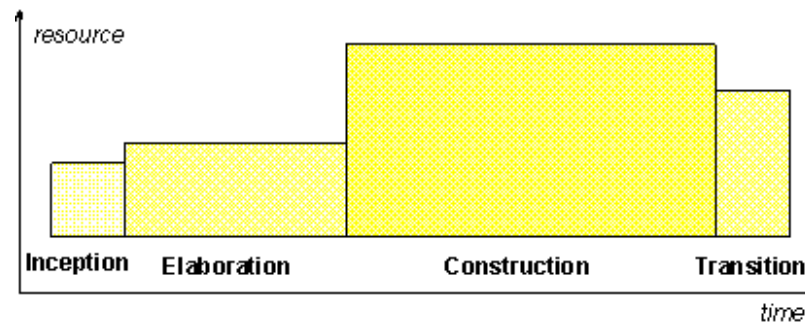


Φάσεις στη RUP: resource histogram



Φάσεις στη RUP

	<u>Inception</u>	<u>Elaboration</u>	<u>Construction</u>	<u>Transition</u>
Effort	~5 %	20 %	65 %	10%
Schedule	10 %	30 %	50 %	10%



Σύλληψη της ιδέας

Περίγραμμα

- Καθορισμός των στόχων του κύκλου ζωής του έργου
- Μελέτη (consideration) των αναγκών των εμπλεκομένων (stakeholders)
- Εμβέλεια, οριακές συνθήκες, κριτήρια αποδοχής
- Καθορισμός μέρους των απαιτήσεων

Κριτήρια εισόδου: διατύπωση μιας ανάγκης

- Ένα όραμα
- Ένα υπάρχον (legacy) σύστημα με απαιτούμενες βελτιώσεις
- Ένας διαγωνισμός προμήθειας
- Κάποιοι πόροι (resources)
- Ένα κάποιο prototype "mock-up"

Σύλληψη της ιδέας: δραστηριότητες

Εμβέλεια του έργου

- Ανάγκες για κάθε stakeholder
- Κριτήρια αποδοχής

Καθορισμός του business case

- Στρατηγική διαχείρισης ρίσκου, αρχικό πλάνο του έργου,
- Κόστος, χρονικός προγραμματισμός

Καθορισμός μιας υποψήφιας αρχιτεκτονικής

Προετοιμασία του περιβάλλοντος ανάπτυξης του έργου

Σύλληψη της ιδέας: ολοκλήρωση

Υπάρχει κάποιο διατυπωμένο business case με περιγραφή ενός οράματος για το λογισμικό σε όρους λειτουργικότητας, εμβέλειας, επιδόσεων, ικανότητας (capacity), τεχνολογιών

Υπάρχουν διατυπωμένα κριτήρια επιτυχίας (πχ αναμενόμενο κέρδος)

Αξιολόγηση και εκτίμηση ρίσκου (αρχική)

Εκτίμηση των πόρων που απαιτούνται για την ολοκλήρωση της επόμενης φάσης επεξεργασίας (elaboration phase)

Επεξεργασία

Περίγραμμα

- Καθορισμός κινδύνων
- Διατύπωση "ευσταθούς" οράματος για το προϊόν
- Καθορισμός αρχιτεκτονικής
- Προγραμματισμός απαιτούμενων πόρων – κόστους

Κριτήρια εισόδου

- Η πλήρης διαθεσιμότητα των αποτελεσμάτων της προηγούμενης φάσης
- Ένα εγκεκριμένο πλάνο και προϋπολογισμός του έργου
- Οι πόροι που απαιτούνται

Επεξεργασία: δραστηριότητες

Καθορισμός της αρχιτεκτονικής

- Πλάνο έργων
- Διαδικασίες, απαιτούμενες υποδομές και περιβάλλον ανάπτυξης

Επικύρωση της αρχιτεκτονικής (validation)

Ορισμός αρχιτεκτονικής αναφοράς (baseline)

Επεξεργασία: ολοκλήρωση

Ενα λεπτομερές πλάνο ανάπτυξης λογισμικού

- Εκτίμηση κινδύνων, πλάνο διαχείρισης και στελέχωσης, καθορισμός των φάσεων και των επαναλήψεων, πλάνο ελέγχου

Το περιβάλλον και τα εργαλεία ανάπτυξης

Ενα σύνολο από κριτήρια αξιολόγησης του τελικού προϊόντος

Ενα μοντέλο ανάλυσης του θεματικού πεδίου, λεπτομερές όσο απαιτείται για τον έλεγχο πληρότητας της αρχιτεκτονικής

Μια εφαρμόσιμη αρχιτεκτονική αναφοράς

Κατασκευή

Περίγραμμα

- Διαχείριση των διατιθέμενων πόρων για την εκτέλεση του έργου ανάπτυξης λογισμικού εντός χρόνου, προϋπολογισμού και όρων ποιότητας
- Διαίρεση σε επαναλήψεις (iterations)

Κριτήρια εισόδου

- Τα προϊόντα της (κάθε) προηγούμενης επανάληψης
- Το πλάνο της τρέχουσας επανάληψης
- Εκτίμηση ρίσκου για την τρέχουσα επανάληψη
- Προβλήματα που θα αντιμετωπίσει η τρέχουσα επανάληψη

Κατασκευή: δραστηριότητες

Ανάπτυξη συστατικών στοιχείων λογισμικού (components)

- Ικανοποίηση των περιπτώσεων χρήσης, σεναρίων και απαιτούμενης λειτουργικότητας
- Έλεγχος μονάδων και integration

Διαχείριση πόρων και έλεγχος της διαδικασίας

Αξιολόγηση του κύκλου επανάληψης (iteration)

- Ικανοποίηση των στόχων του κύκλου επανάληψης

Κατασκευή: ολοκλήρωση

Τα συστατικά στοιχεία λογισμικού, ενημερωμένα με τα χαρακτηριστικά που (τους) προστέθηκαν κατά τον κύκλο επανάληψης

Η περιγραφή των αποτελεσμάτων του κύκλου επανάληψης

Περιπτώσεις και αποτελέσματα ελέγχων (testing)

Ποσοτικά κριτήρια για την αξιολόγηση των αποτελεσμάτων του/των επόμενου/ων κύκλου/ων επανάληψης

Μετάβαση

Περίγραμμα

- Διάθεση του προϊόντος (λογισμικού) στους χρήστες
- Marketing, διανομή, εγκατάσταση, ρυθμίσεις, υποστήριξη χρηστών, υλοποίηση διορθώσεων, κ.ά.

Κριτήρια εισόδου

- Τα προϊόντα της φάσης κατασκευής σε ώριμη κατάσταση ώστε να φτάσουν στους τελικούς χρήστες

Μετάβαση: δραστηριότητες και ολοκλήρωση

Ελεγχος του τελικού προϊόντος στο περιβάλλον του πελάτη

Προσαρμογές (fine tuning) σύμφωνα με υποδείξεις του πελάτη

Διάθεση στον πελάτη για παραγωγική χρήση

Υλικά υποστήριξης χρηστών

Ολοκλήρωση

- Επικαιροποιημένη τεκμηρίωση του λογισμικού αναφορικά με τους αρχικούς στόχους, επιδιωκόμενα χαρακτηριστικά και μέτρα
- Νέες δυνατότητες του κατασκευαστή μετά την ολοκλήρωση του κύκλου / έργου

RUP: αρχές μοντελοποίησης

Βασικές λειτουργίες

- Business modeling
- Requirements
- Analysis and design
- Implementation
- Testing
- Deployment

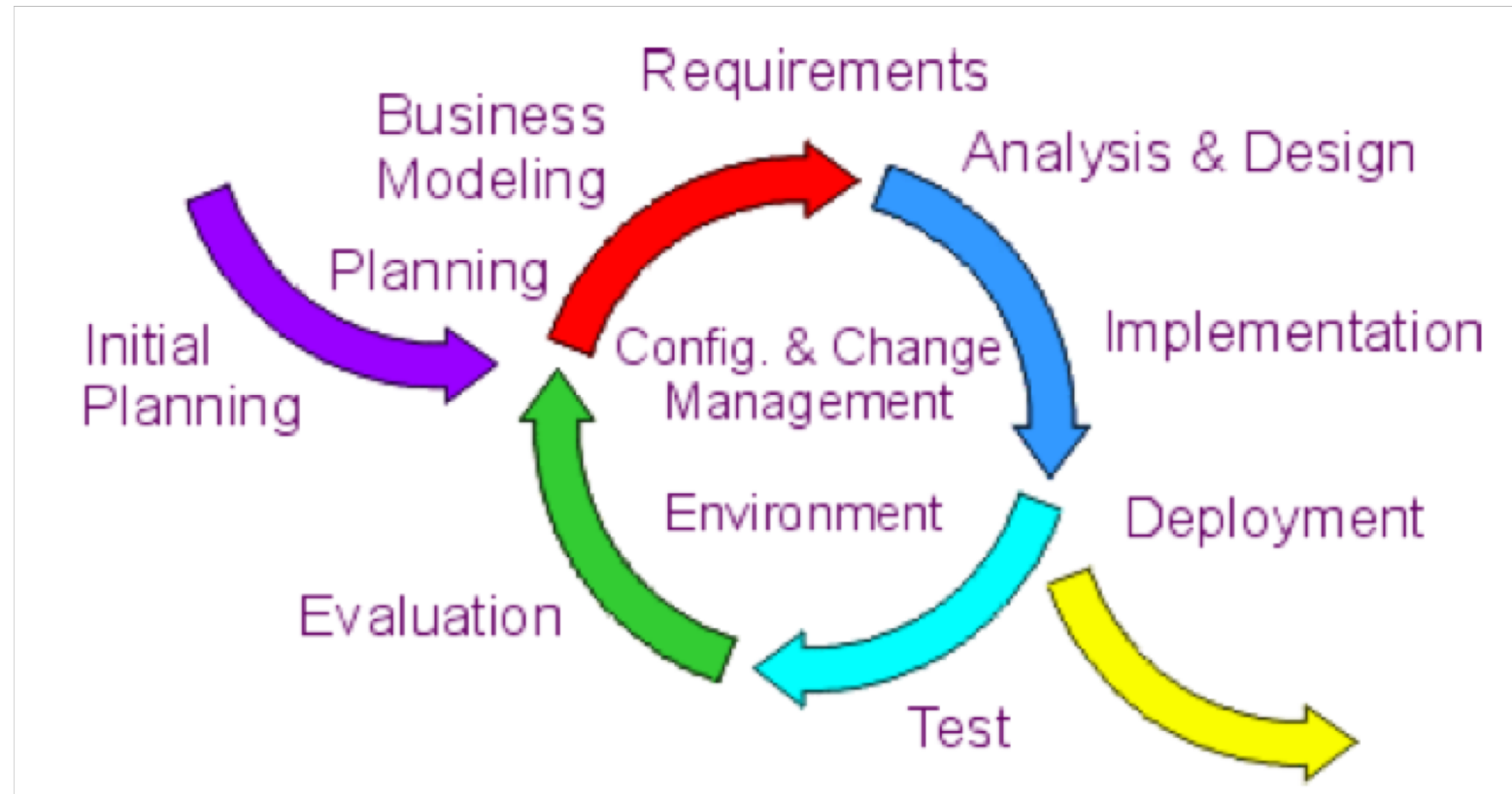
Βοηθητικές λειτουργίες

- Configuration & change management
- Project management
- Environment management and setup

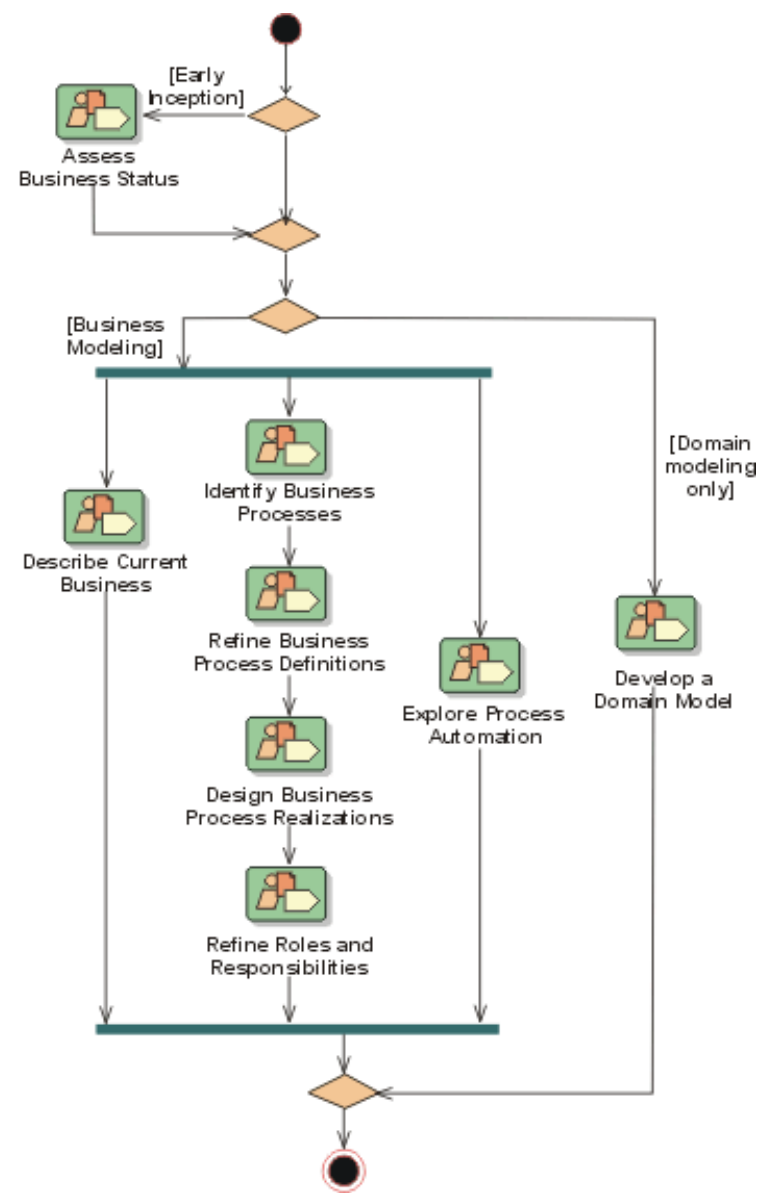
RUP



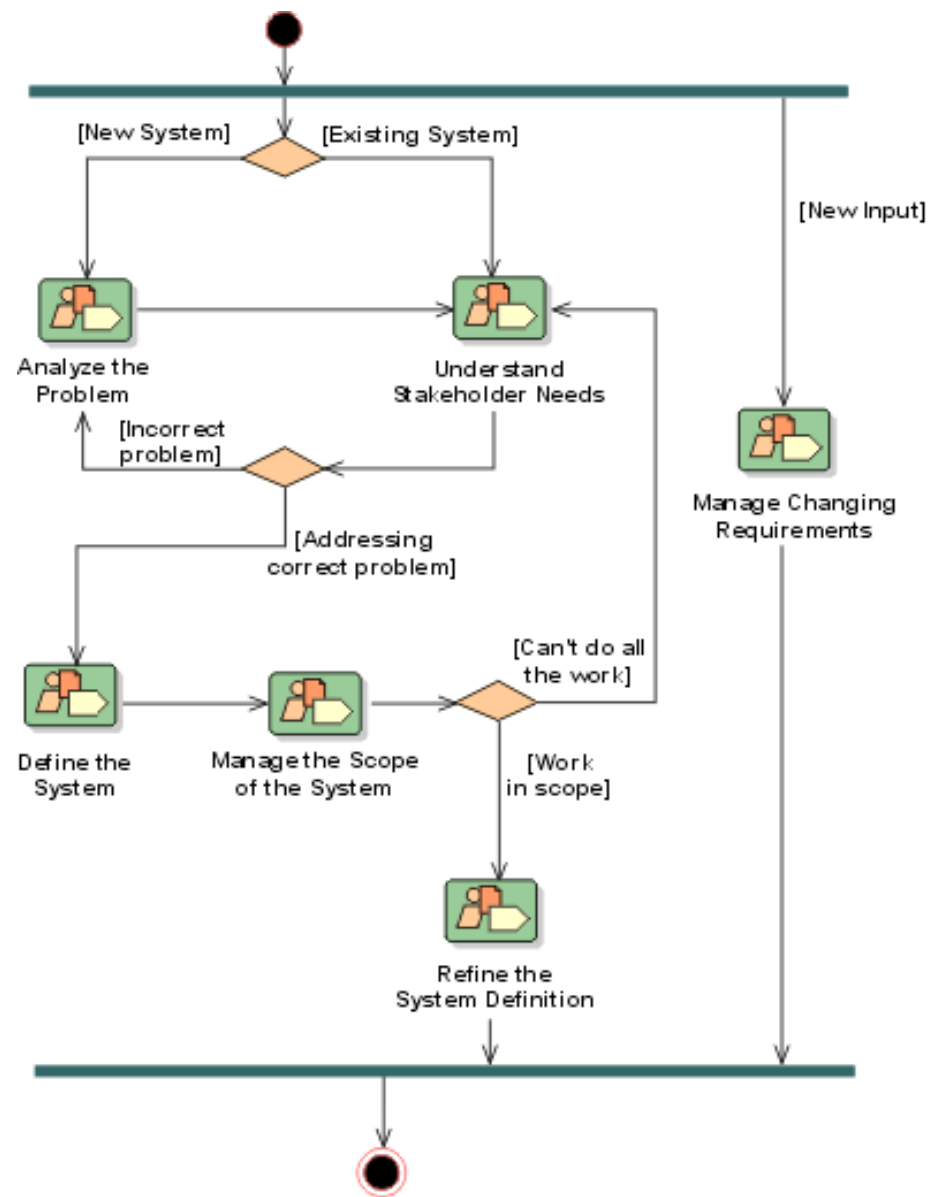
RUP



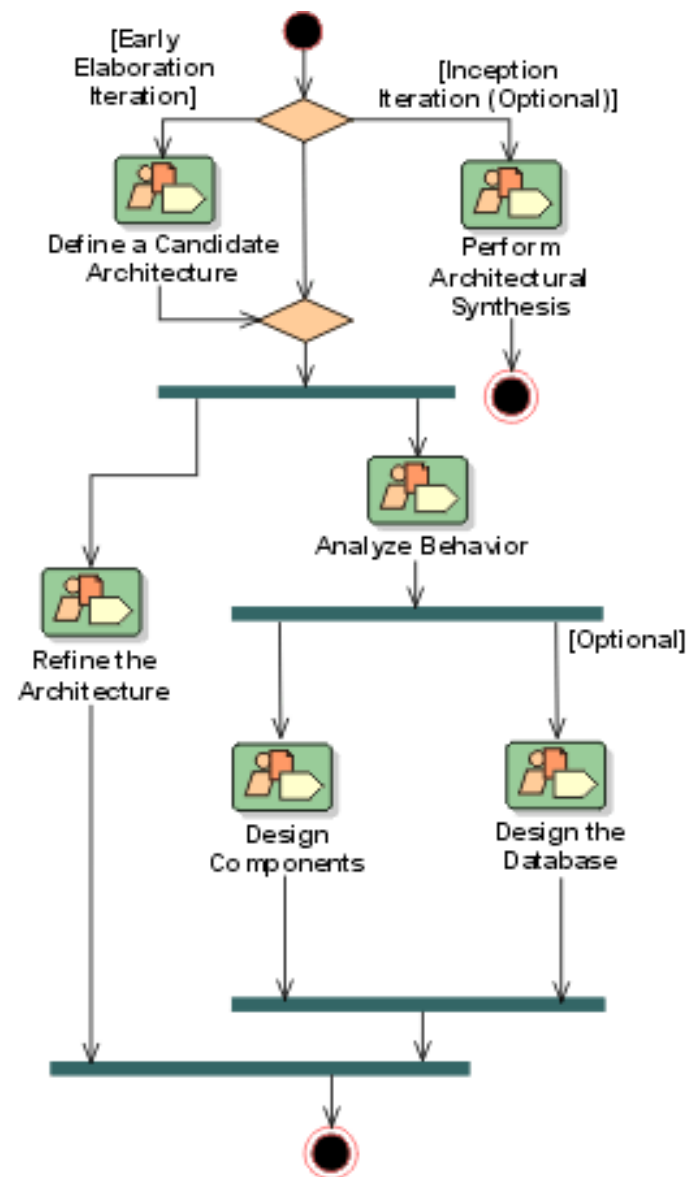
Business modeling



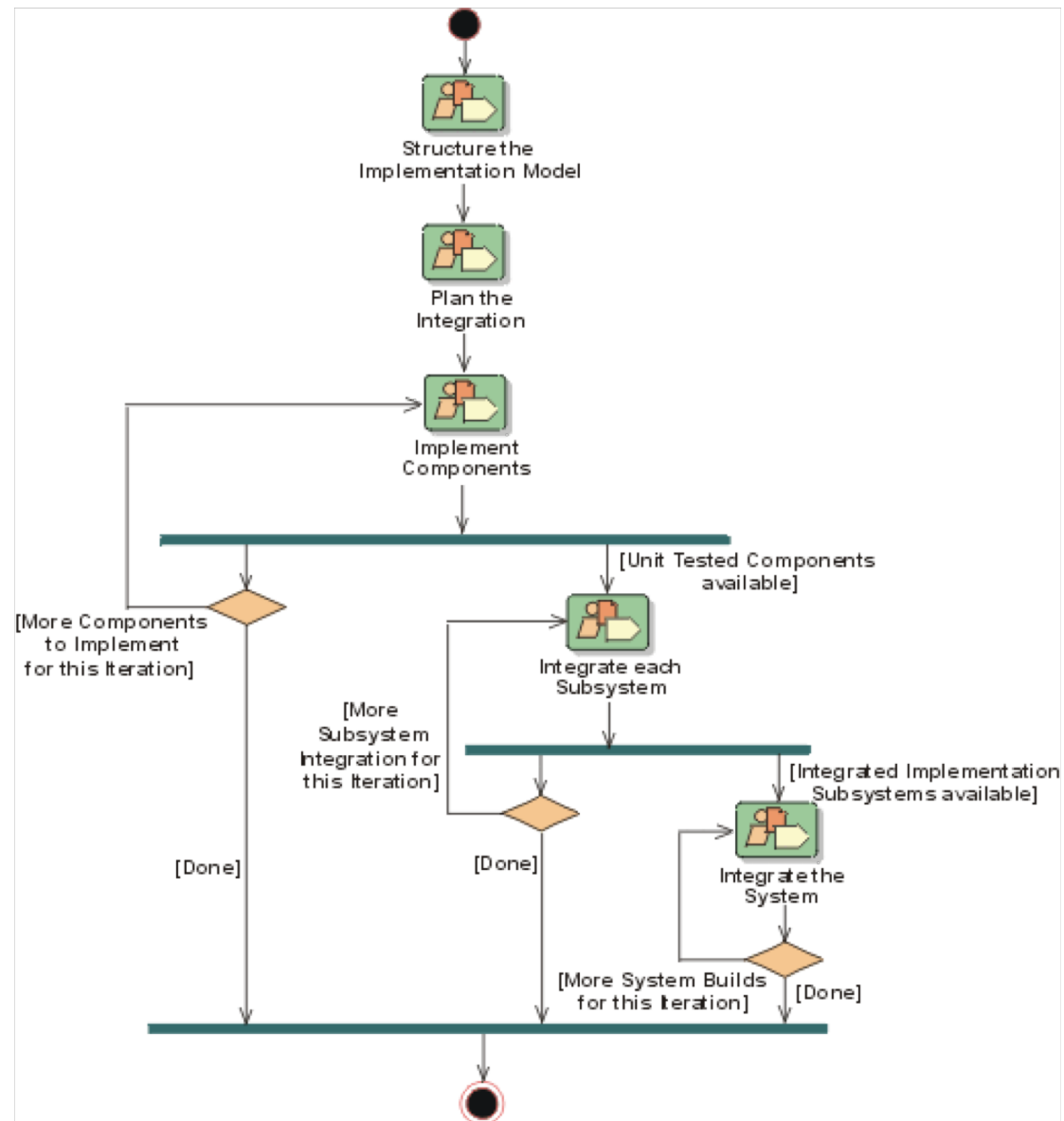
Requirements



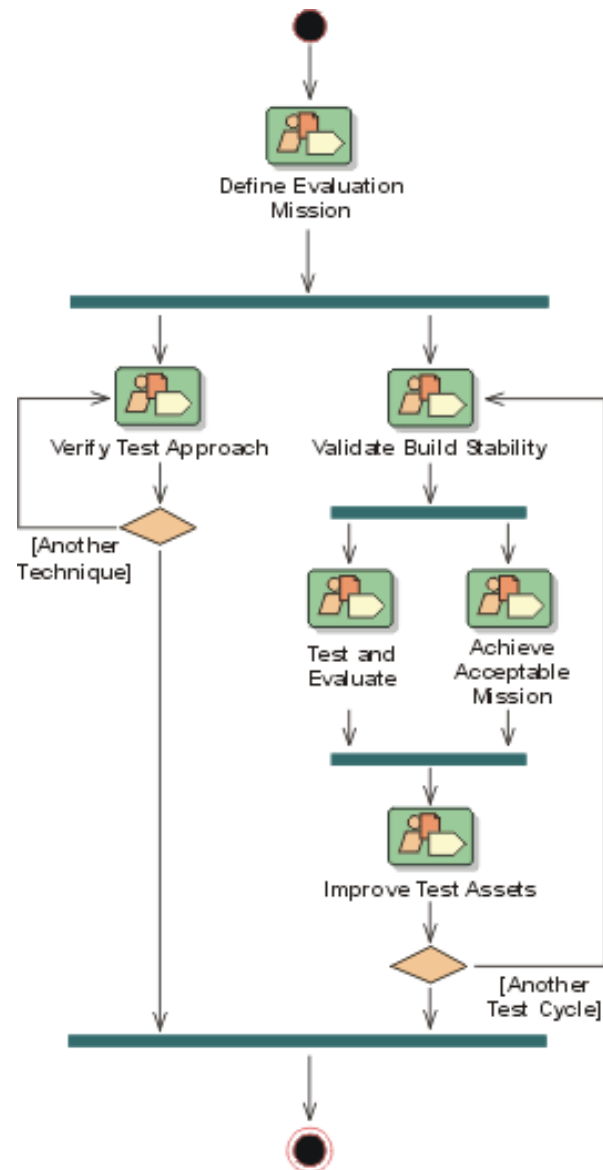
Analysis and design



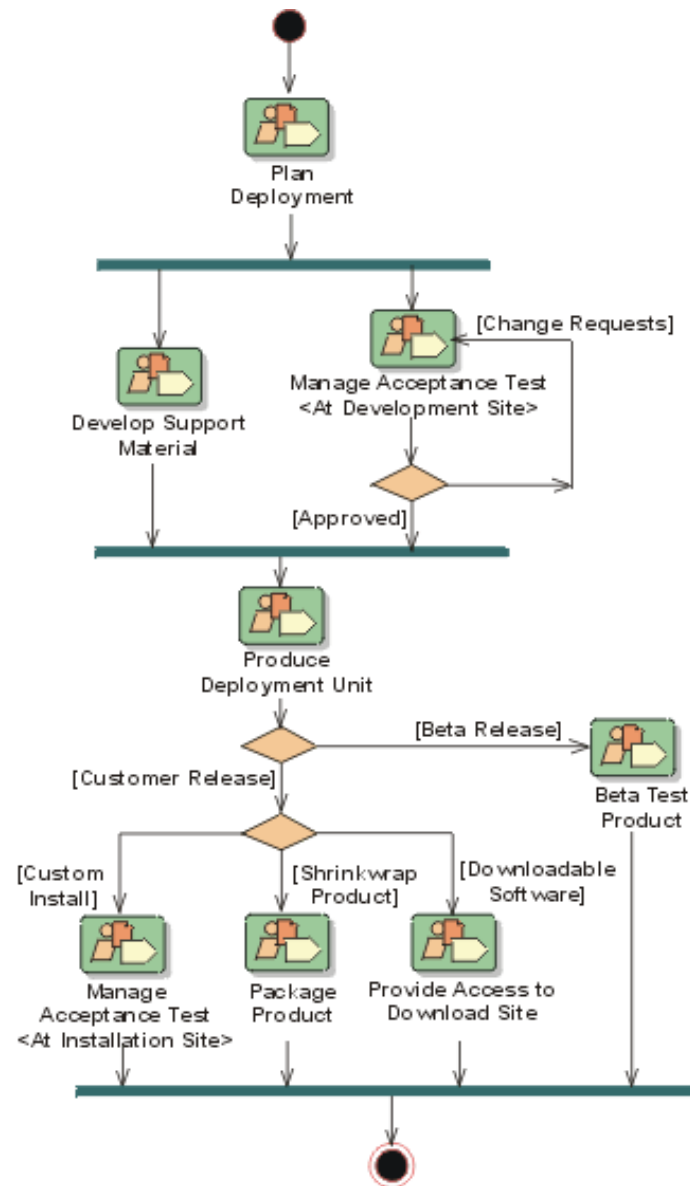
Implementation



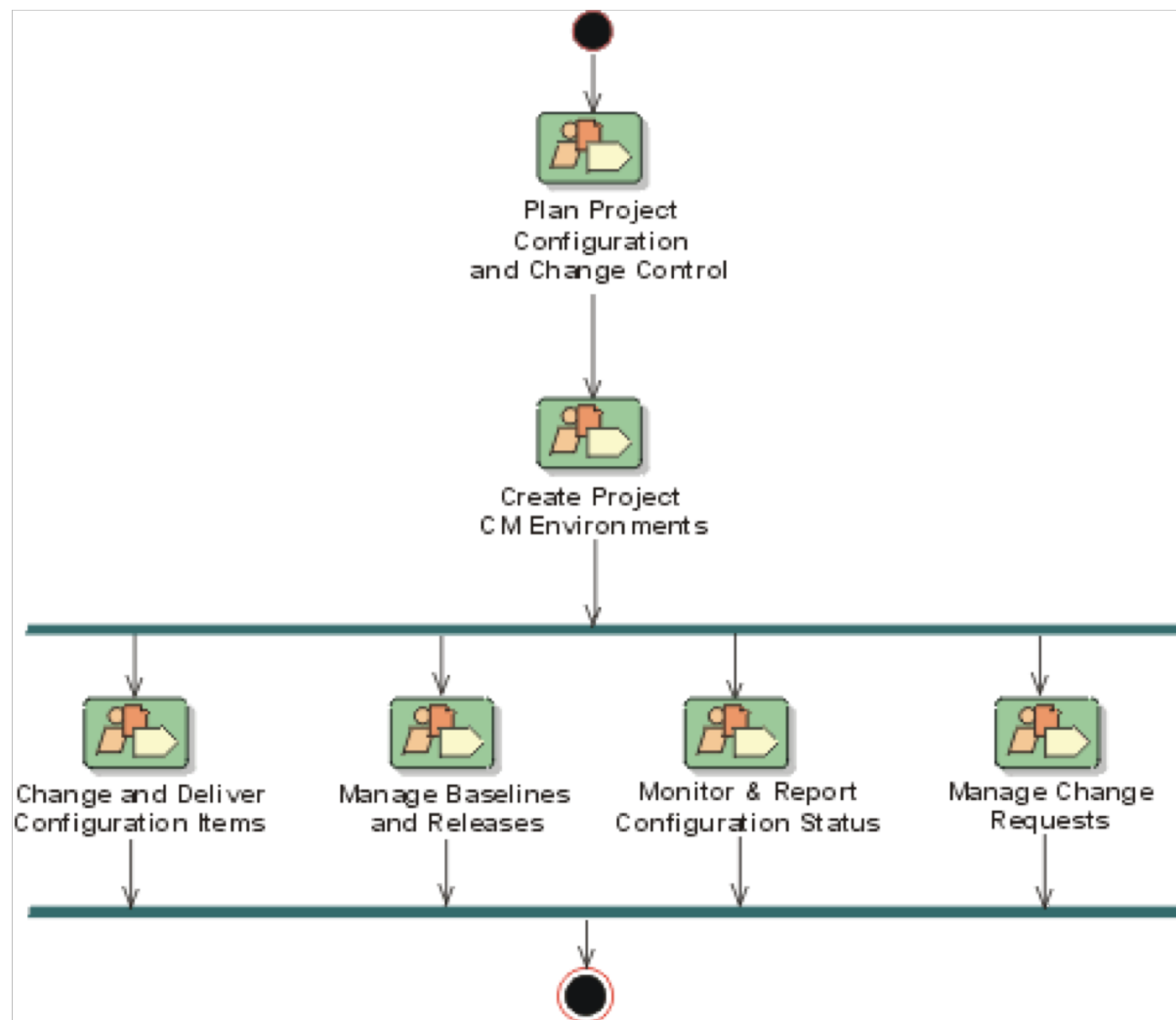
Test



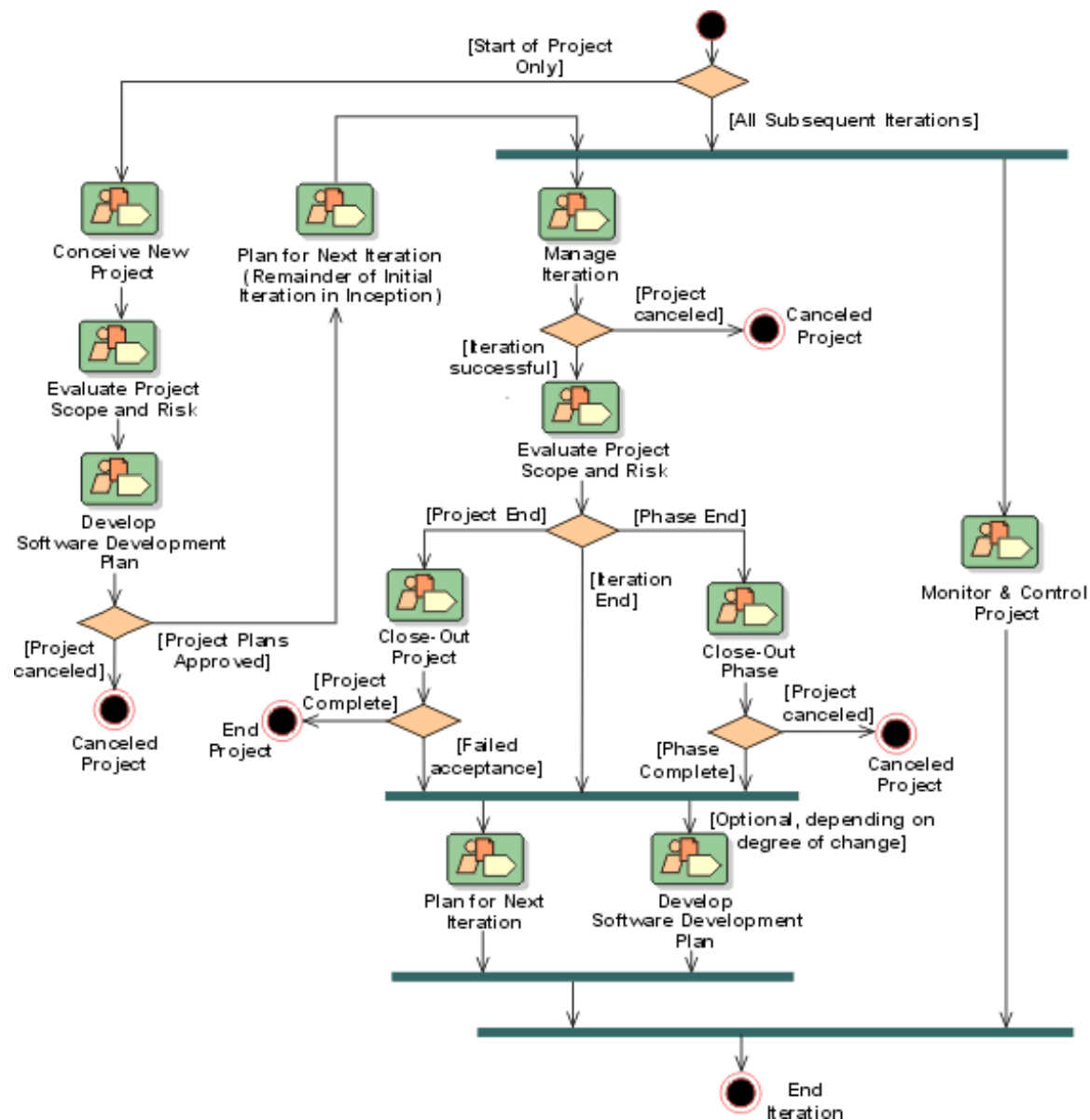
Deployment



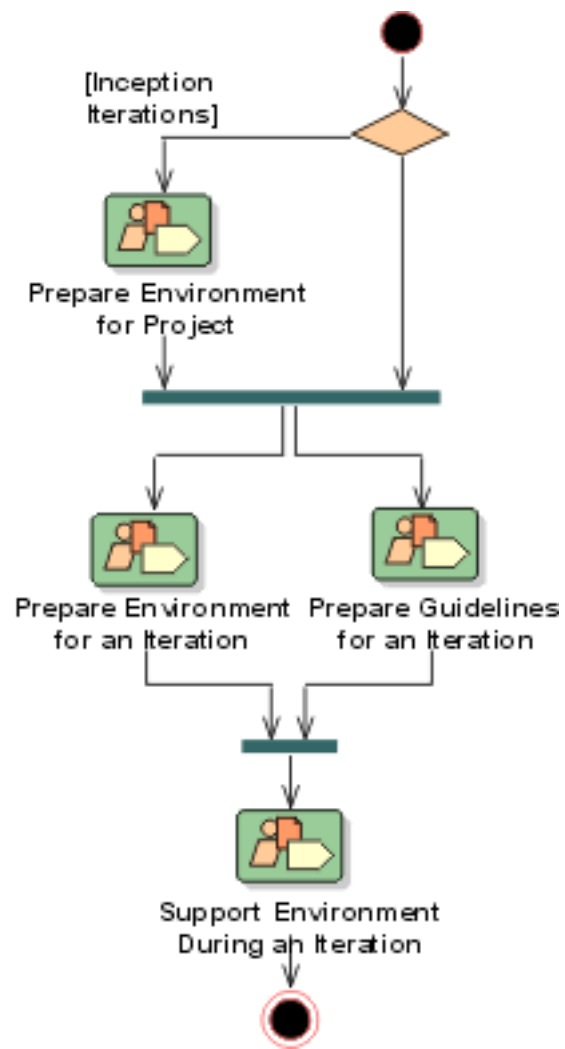
Configuration – change management



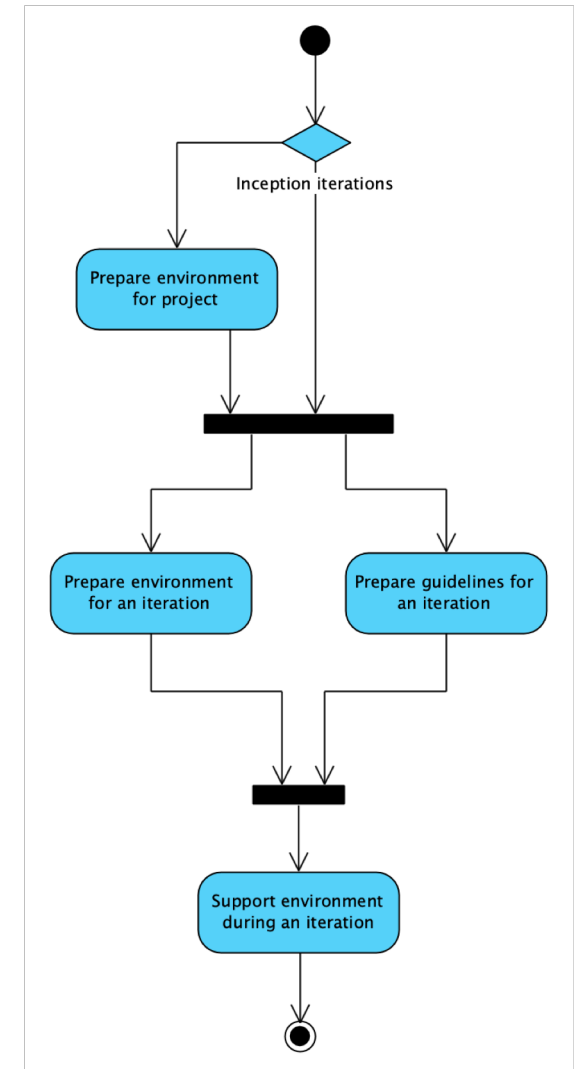
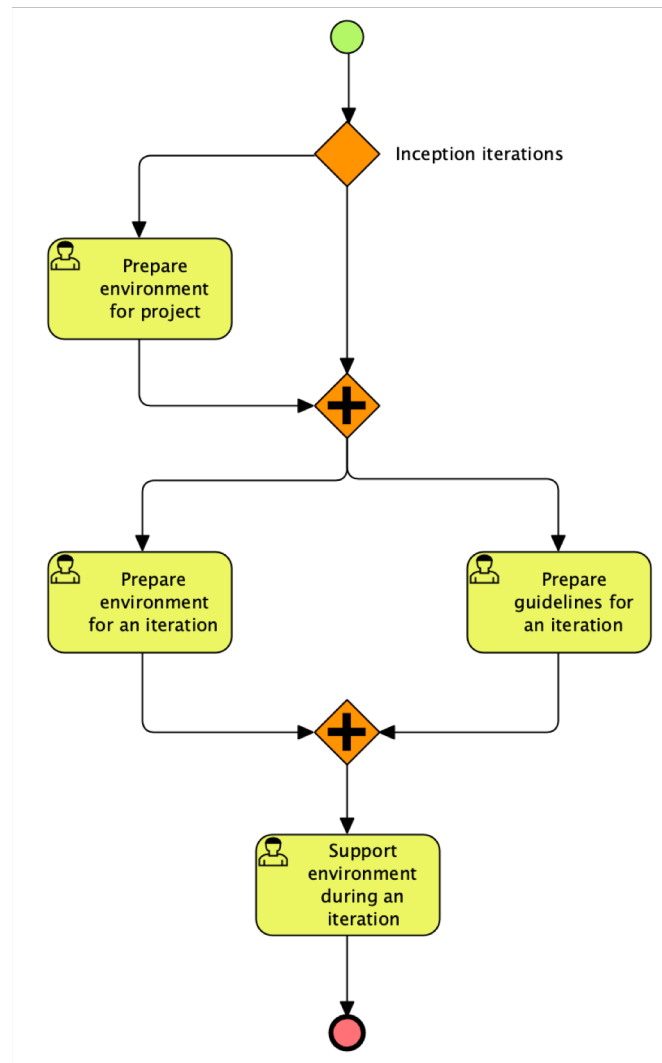
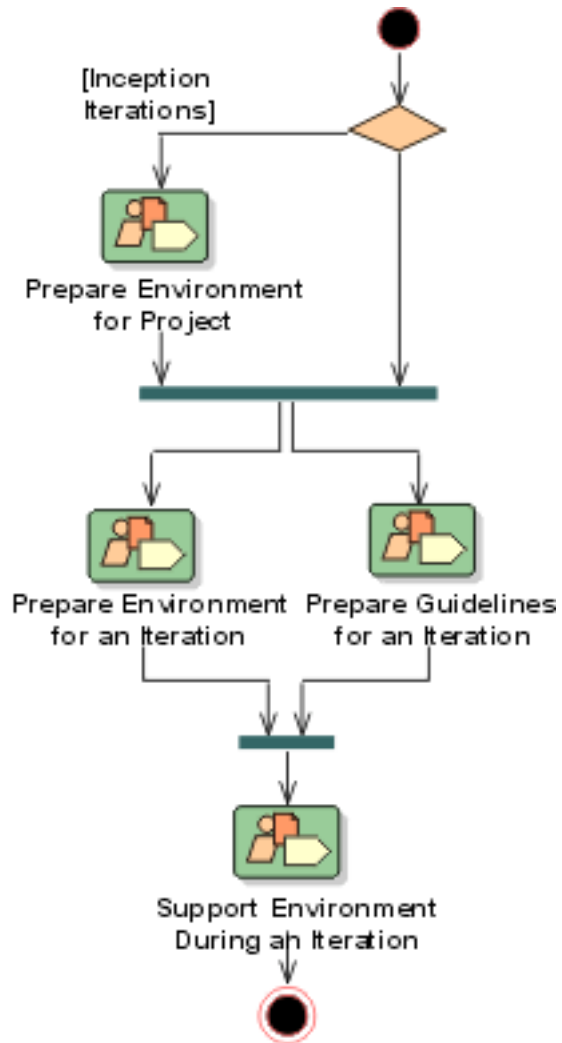
Project management



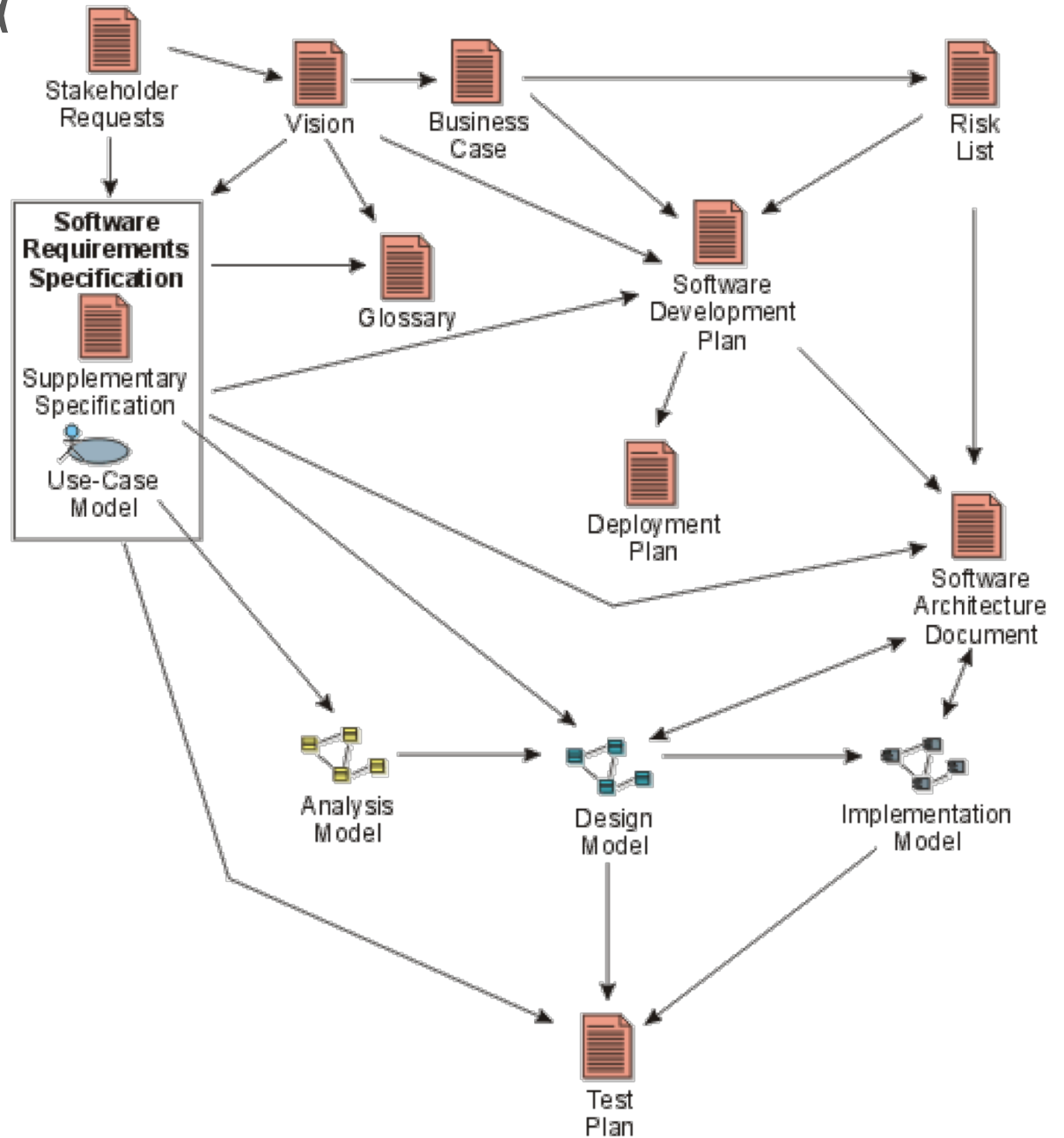
Environment



Notations: BPM, UML



Μοντέλα και έγγραφα



Roles

Analysts

- System Analyst, Business Designer, Business-Model Reviewer, Business-Process Analyst, Requirements Reviewer, Requirements Specifier, Test Analyst, User-Interface Designer

Developers

- Capsule Designer, Code Reviewer, Database Designer, Implementer, Software Architect, Architecture Reviewer, Design Reviewer, Designer, Test Designer

Testers

- Tester

Roles (...)

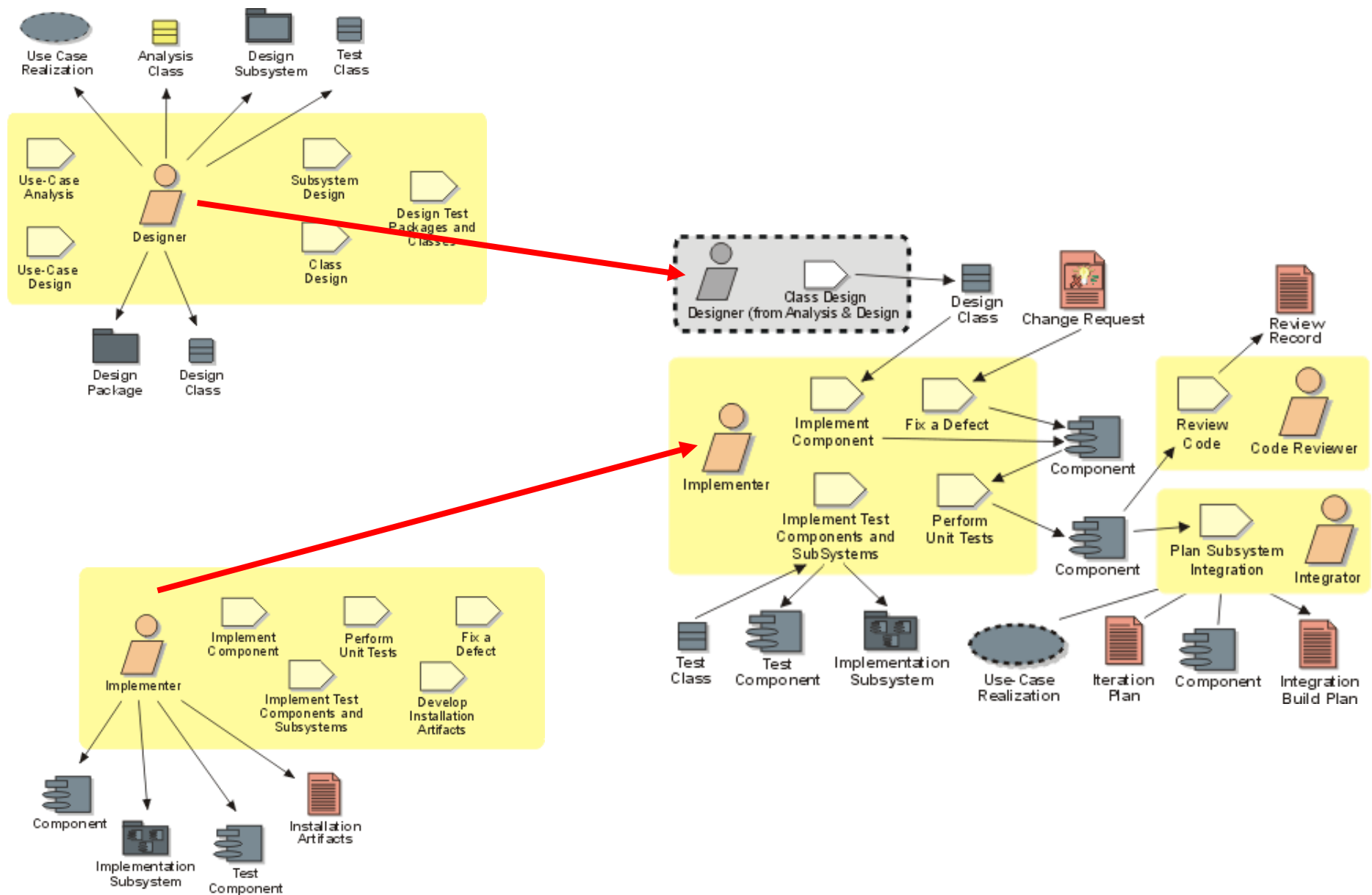
Managers

- Process Engineer, Project Manager, Change Control Manager, Configuration Manager, Deployment Manager, Project Reviewer, Test Manager

Other

- Stakeholder, Course Developer, Graphic Artist, Tool Specialist, System Administrator, Technical Writer

Roles example



Συζήτηση

- Επιτρέπει την έγκαιρη αποτίμηση των κινδύνων
- Επιτρέπει την εξειδίκευση (refinement) των απαιτήσεων κατά τη διάρκεια του έργου [δεν απαιτεί να είναι πλήρως γνωστές στην αρχή]
- Δεν βασίζεται σε κείμενα (documents) ούτε σε τυποποιημένες «τελετουργίες» ανάπτυξης
- Εστιάζει στο ίδιο το λογισμικό και με την έννοια αυτή, υποστηρίζει την ανάπτυξη λογισμικού καλής ποιότητας
- Δεν είναι και ιδιαίτερα ευέλικτη (agile) - μπορεί (;) να προσαρμοστεί
- Δεν ενσωματώνει ξεκάθαρες οδηγίες – κατευθύνσεις για την υλοποίηση
- Απαιτεί πολύ δουλειά στην προσαρμογή στο εκάστοτε περιβάλλον

Agile software development

Introduction and principles

Βασικές αρχές

Τι είναι;

- Agile Software Development is an umbrella term for a set of methods and practices based on the values and principles expressed in the Agile Manifesto.
- Solutions evolve through collaboration between self-organizing, cross-functional teams utilizing the appropriate practices for their context.

Αρχές

- Πολυ-λειτουργικές και αυτο-διοικούμενες ομάδες (self-organizing & cross-functional).
- Προσαρμοστικός σχεδιασμός, εξελικτική ανάπτυξη, ταχεία παράδοση και συνεχής βελτίωση.
- Ταχεία και ευέλικτη προσαρμογή στις αλλαγές.

Agile manifesto

1

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4

Business people and developers must work together daily throughout the project.

5

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7

Working software is the primary measure of progress.

8

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9

Continuous attention to technical excellence and good design enhances agility.

10

Simplicity--the art of maximizing the amount of work not done--is essential.

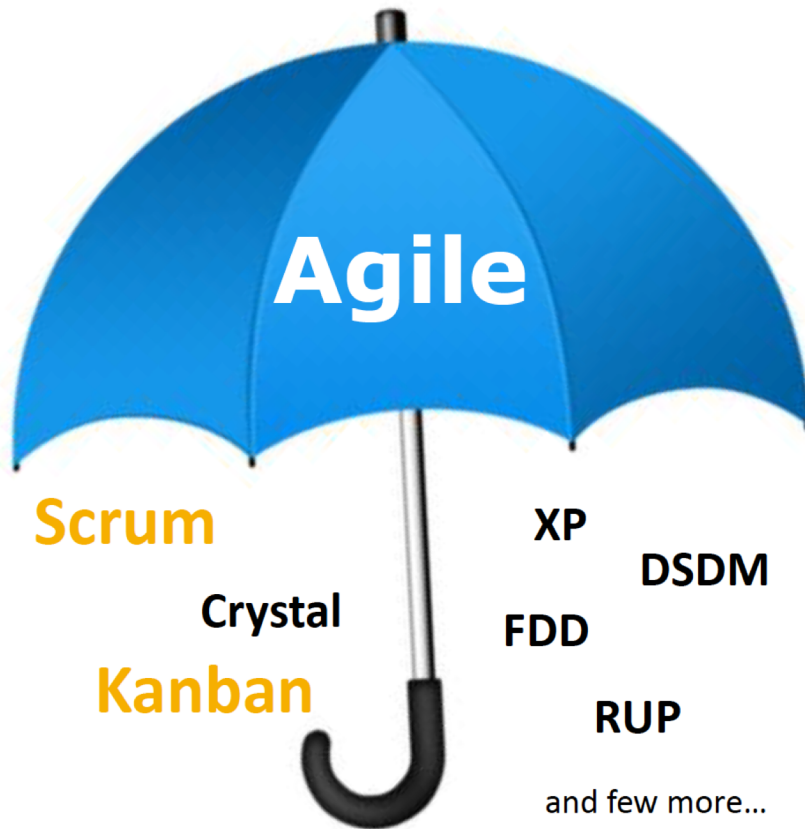
11

The best architectures, requirements, and designs emerge from self-organizing teams.

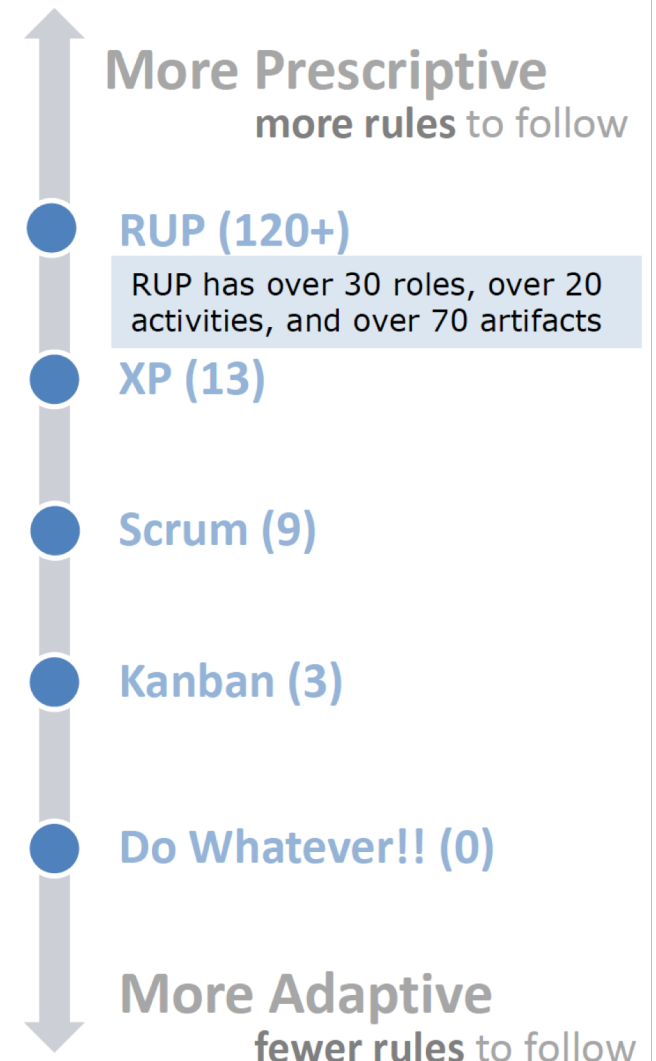
12

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

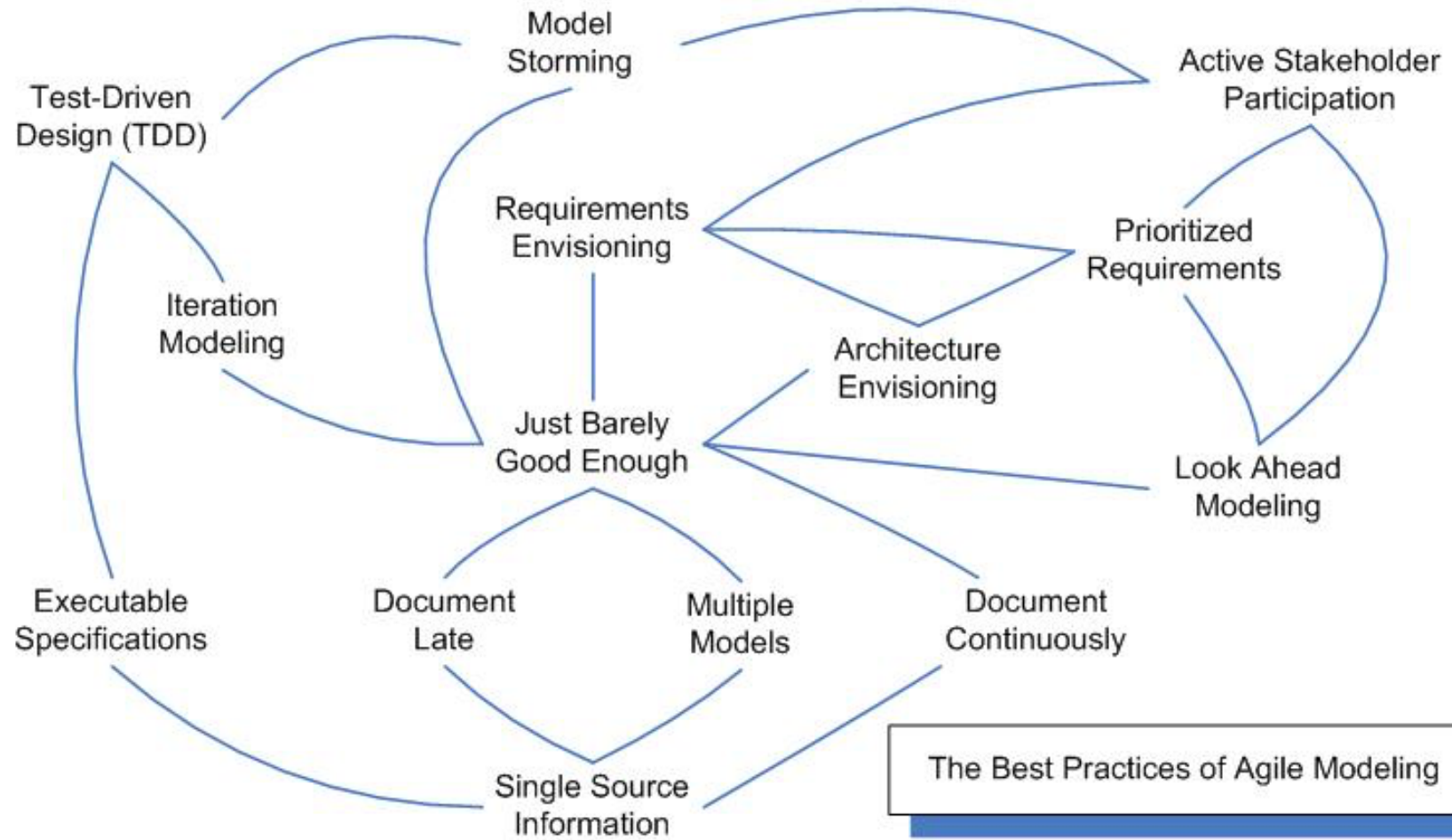
Agile – context



* Check wikipedia for list of all Agile methods

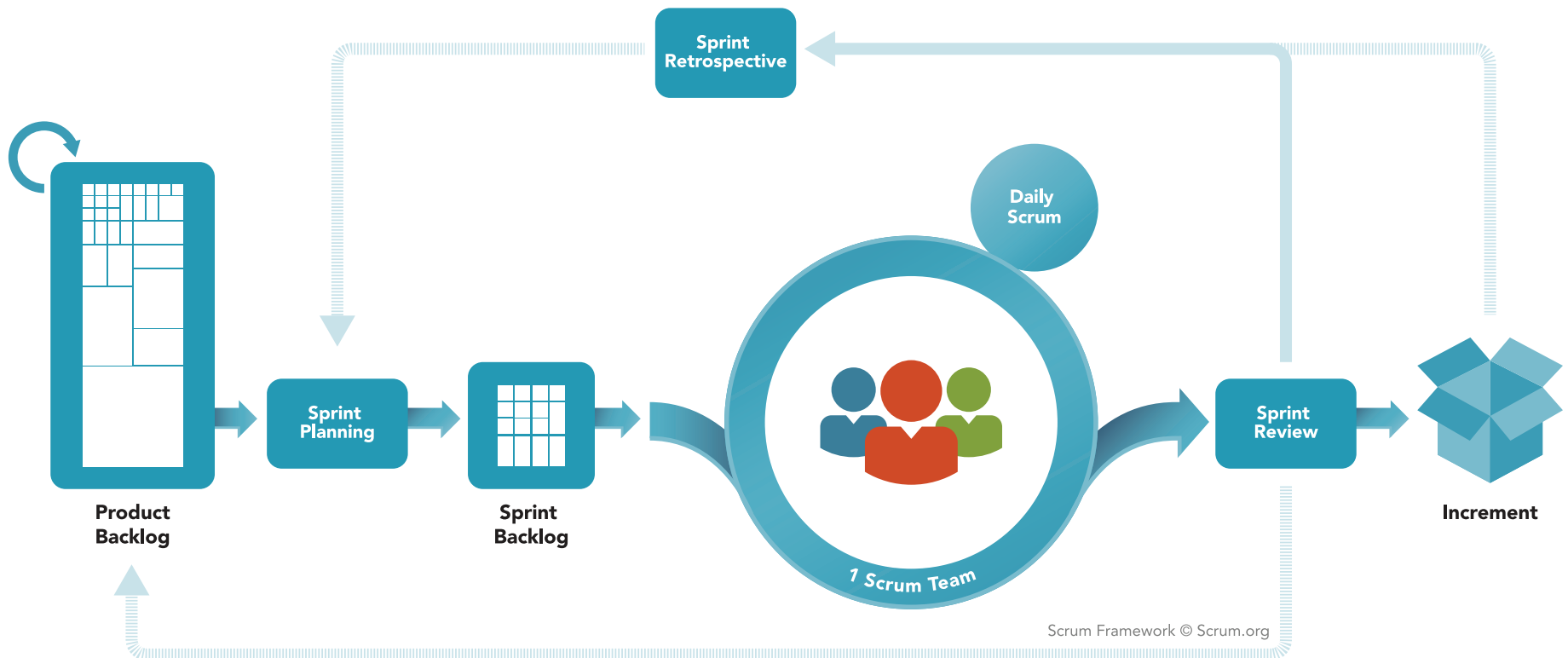


Agile modeling



SCRUM

...Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value... (www.scrum.org)



SCRUM values



SCRUM – βήματα

Sprint: a time-box during which a useable, potentially releasable product increment is created. Sprints have consistent durations in a project.

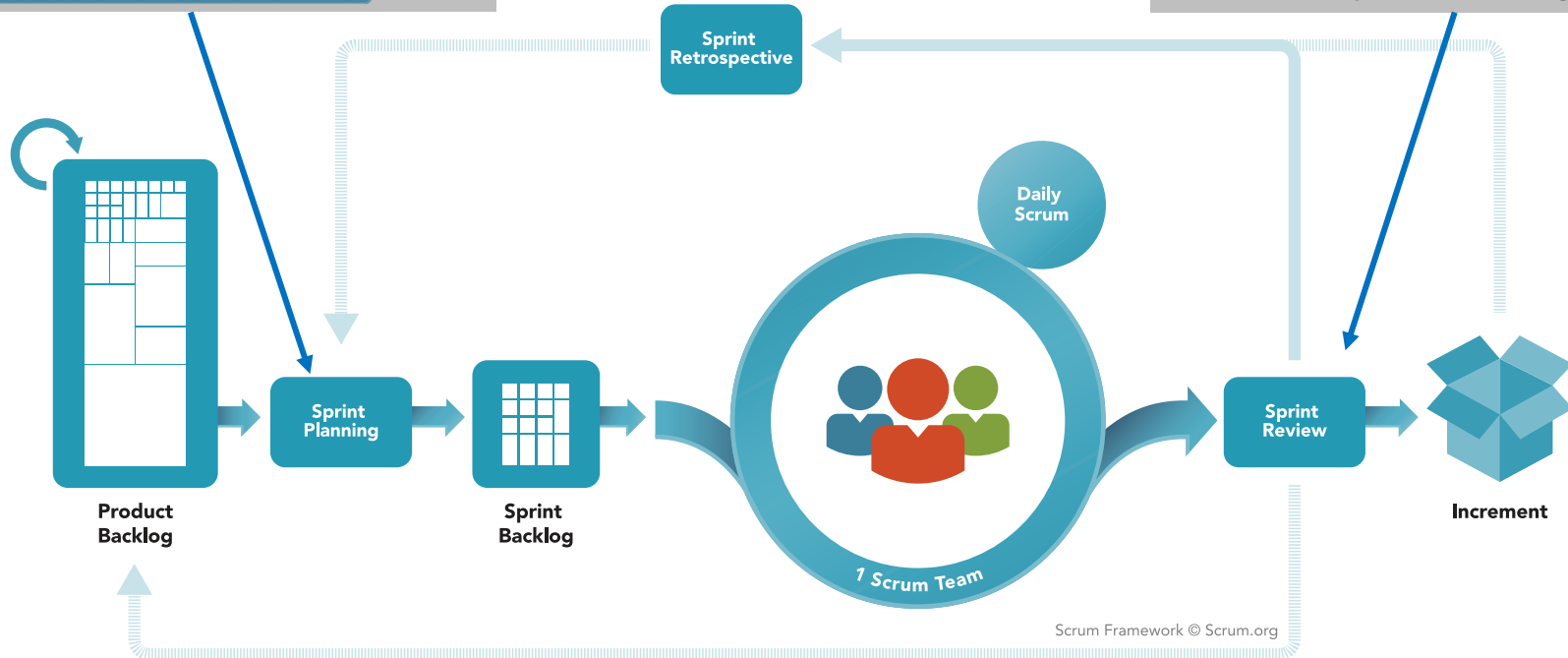
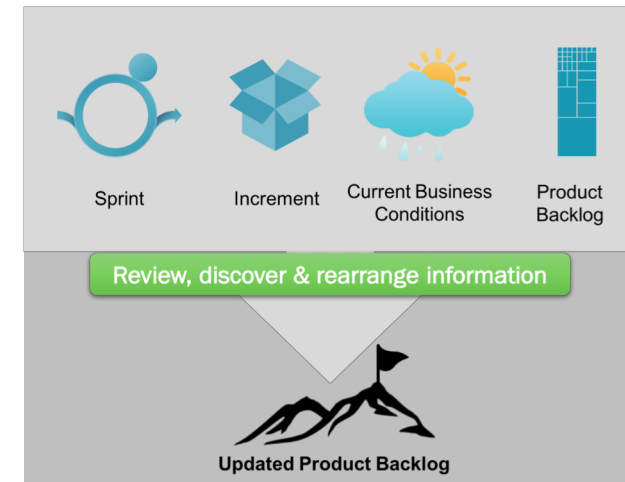
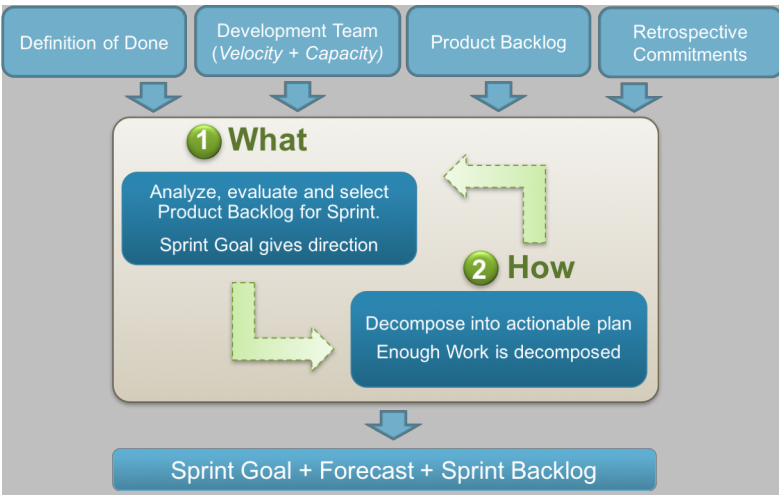
Sprint planning: the work to be performed in the Sprint is planned collaboratively by the entire Scrum Team.

Daily Scrum: a 15-minute time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours

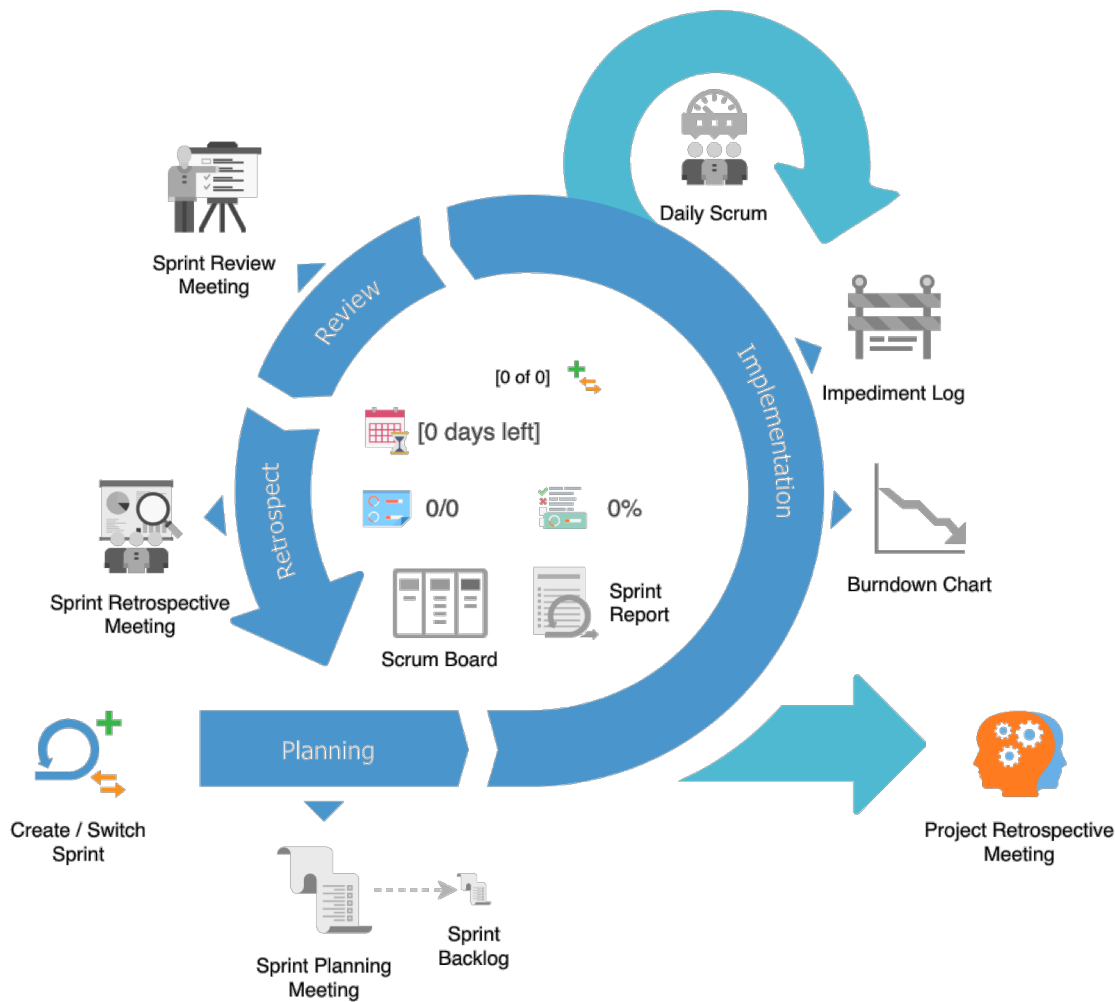
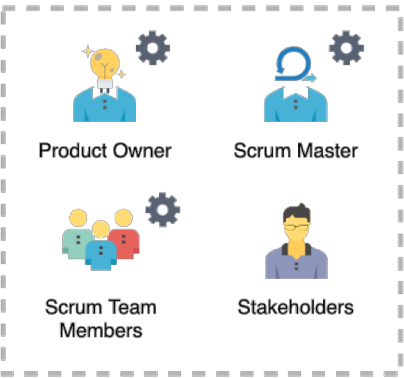
Sprint review: a Sprint Review is held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed

Sprint retrospective: occurs after the Sprint Review and prior to the next Sprint Planning. The Scrum Team inspects itself and creates a plan for improvements for the next Sprint.

SCRUM - βήματα

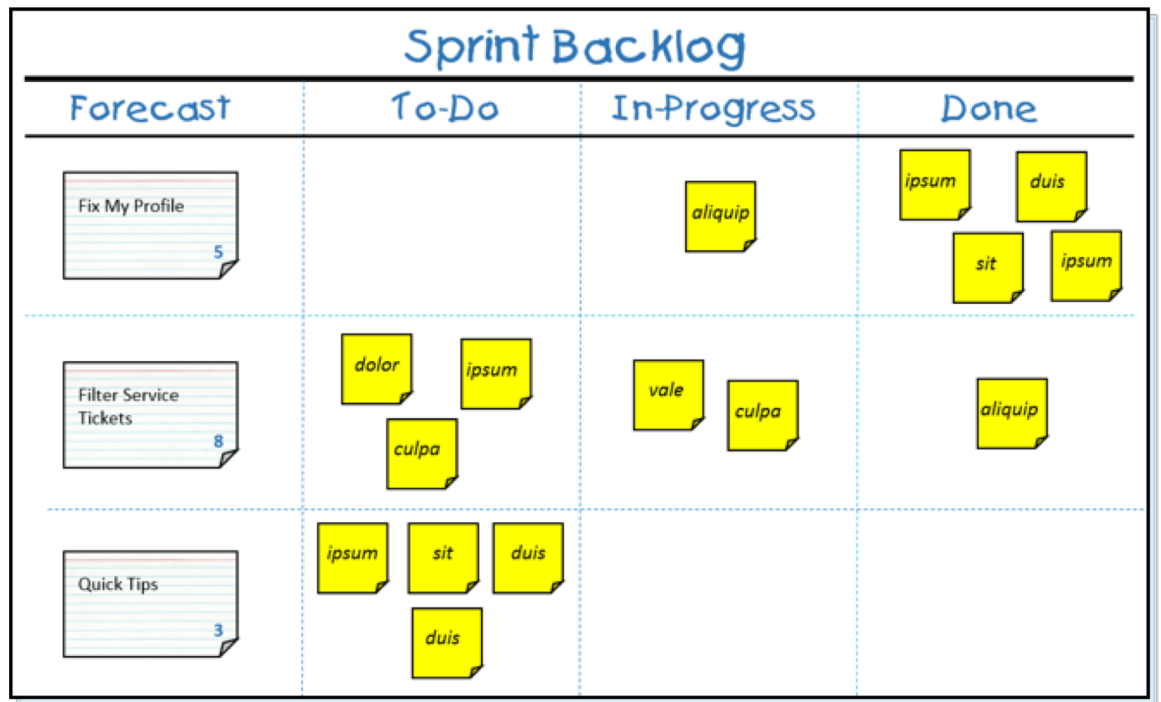
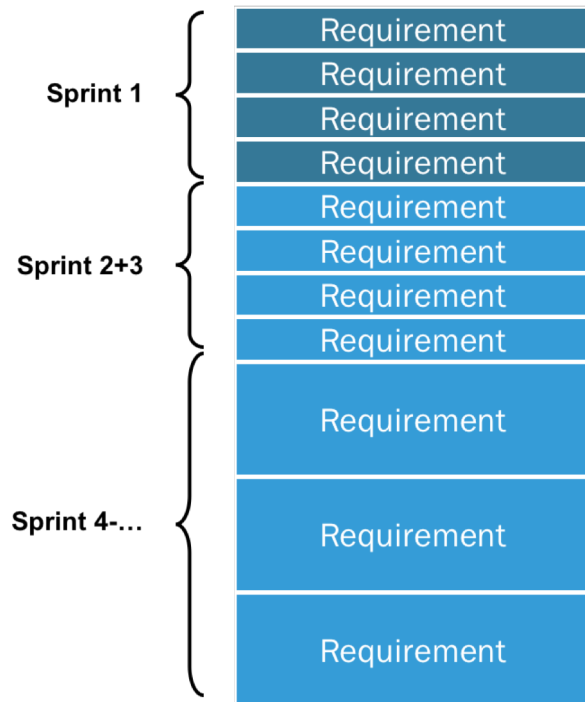


Agile/SCRUM in Software Development



Backlog

Backlog = features to be implemented -> “user stories”



References

Overview of the Rational Unified Process

sce.uhcl.edu/helm/rationalunifiedprocess/process/ovu_proc.htm

Agile modeling and the Rational Unified Process

www.agilemodeling.com/essays/agileModelingRUP.htm#AdoptingAM

Rational Unified Process Best Practices for Software Dev Teams.

www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

SCRUM

www.scrum.org

Discussion