



Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής & Τηλεπικοινωνιών

Τεχνολογία Λογισμικού

8ο Εξάμηνο 2019-20

Μοντέλα & μεθοδολογίες ανάπτυξης λογισμικού

Δρ. Κώστας Σαΐδης (saiko@di.uoa.gr)

Κύκλος ζωής του λογισμικού

- Συγκέντρωση, καταγραφή και ανάλυση απαιτήσεων λογισμικού.
- Σχεδιασμός λογισμικού.
- Υλοποίηση λογισμικού.
- Έλεγχος, επαλήθευση και επικύρωση λογισμικού.
- Εγκατάσταση, έλεγχος, παραμετροποίηση και ολοκλήρωση λογισμικού στο παραγωγικό του περιβάλλον.
- Συντήρηση και επέκταση λογισμικού.

Τρόπος εκτέλεσης των βημάτων

- Ακολουθιακός / παράλληλος
- Άπαξ / σε επαναλήψεις
- Συχνότητα αλληλεπίδρασης με τους χρήστες

Πολλά μοντέλα & μεθοδολογίες

- Μοντέλο καταρράκτη
- Δομημένη ανάλυση και σχεδιασμός
- Αυξητικό μοντέλο
- Επαναληπτικό μοντέλο
- Πρωτοτυποποίηση
- Rational Unified Process
- Ευέλικτη διαδικασία (agile)
- Scrum
- DevOps

Επίσης

- Σπειροειδές μοντέλο
- Extreme programming (επηρέασε το agile)
- Test-driven development (έμφαση στο testing)
- Model-driven development (έμφαση στο domain-model)
- Behavior-driven development (συνδυασμός των δύο παραπάνω)

Μεθολογίες ανάπτυξης λογισμικού

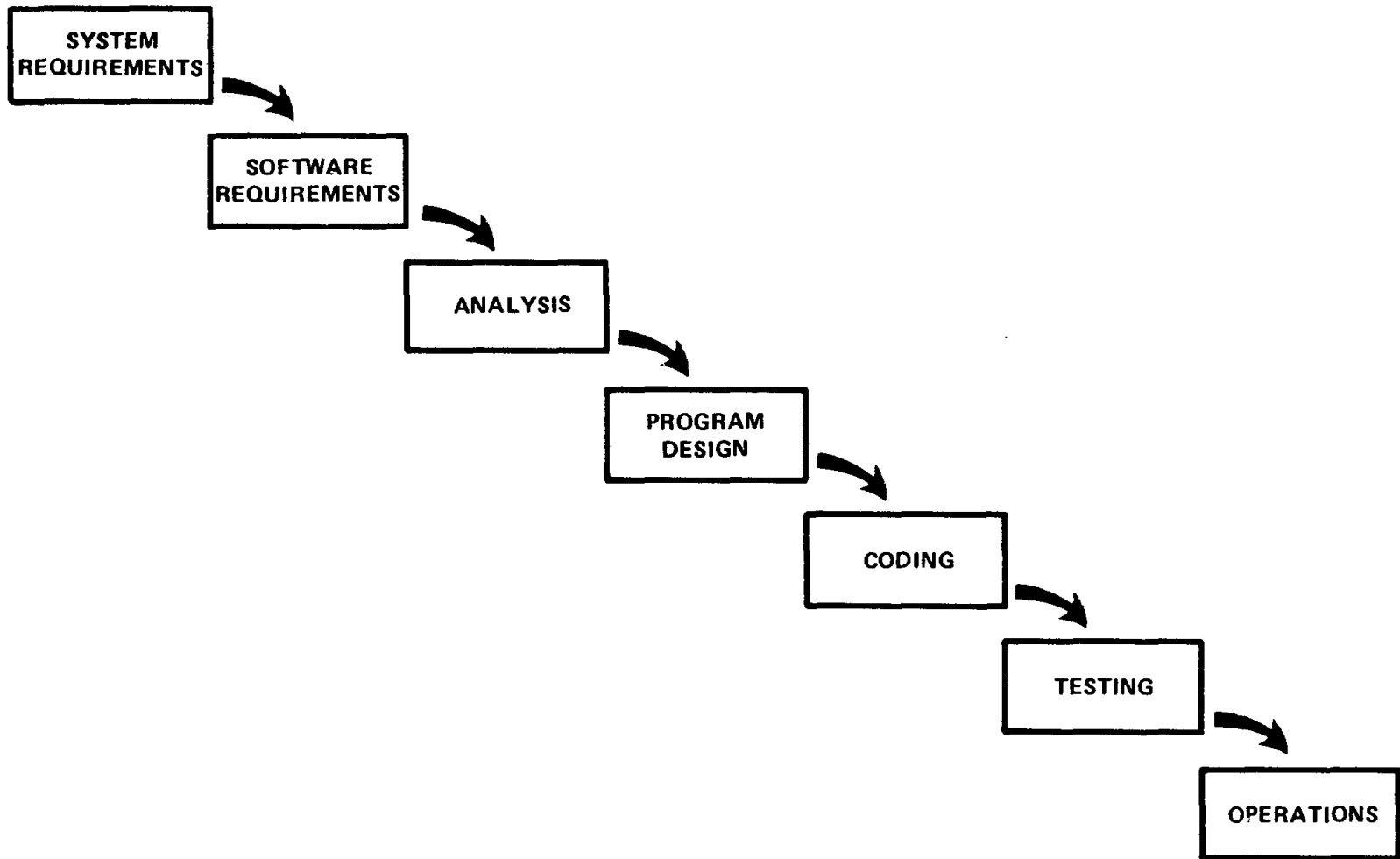
Που διαφέρουν

- Στα εργαλεία ανάλυσης, μοντελοποίησης και σχεδιασμού
- Στον τρόπο εκτέλεσης των βημάτων του κύκλου ζωής
- Στο αν «στοχεύουν» σε συγκεκριμένα είδη συστημάτων (π.χ. Services vs. Products vs. Projects)
- Τι παράγεται (αποτελέσματα, παραδοτέα) και πότε

Σε σύνθετα έργα μπορεί να συνδυάζονται

- π.χ. Agile + Prototyping
- π.χ. Scrum + DevOps

Μοντέλο καταρράκτη



Winston W. Royce

- "Winston Walker Royce (August 15, 1929 – June 7, 1995) was an American computer scientist, director at Lockheed Software Technology Center in Austin, Texas. He was a pioneer in the field of software development, known for his 1970 paper from which the Waterfall model for software development was mistakenly drawn." (Wikipedia).

Λίστα αναγνωσμάτων

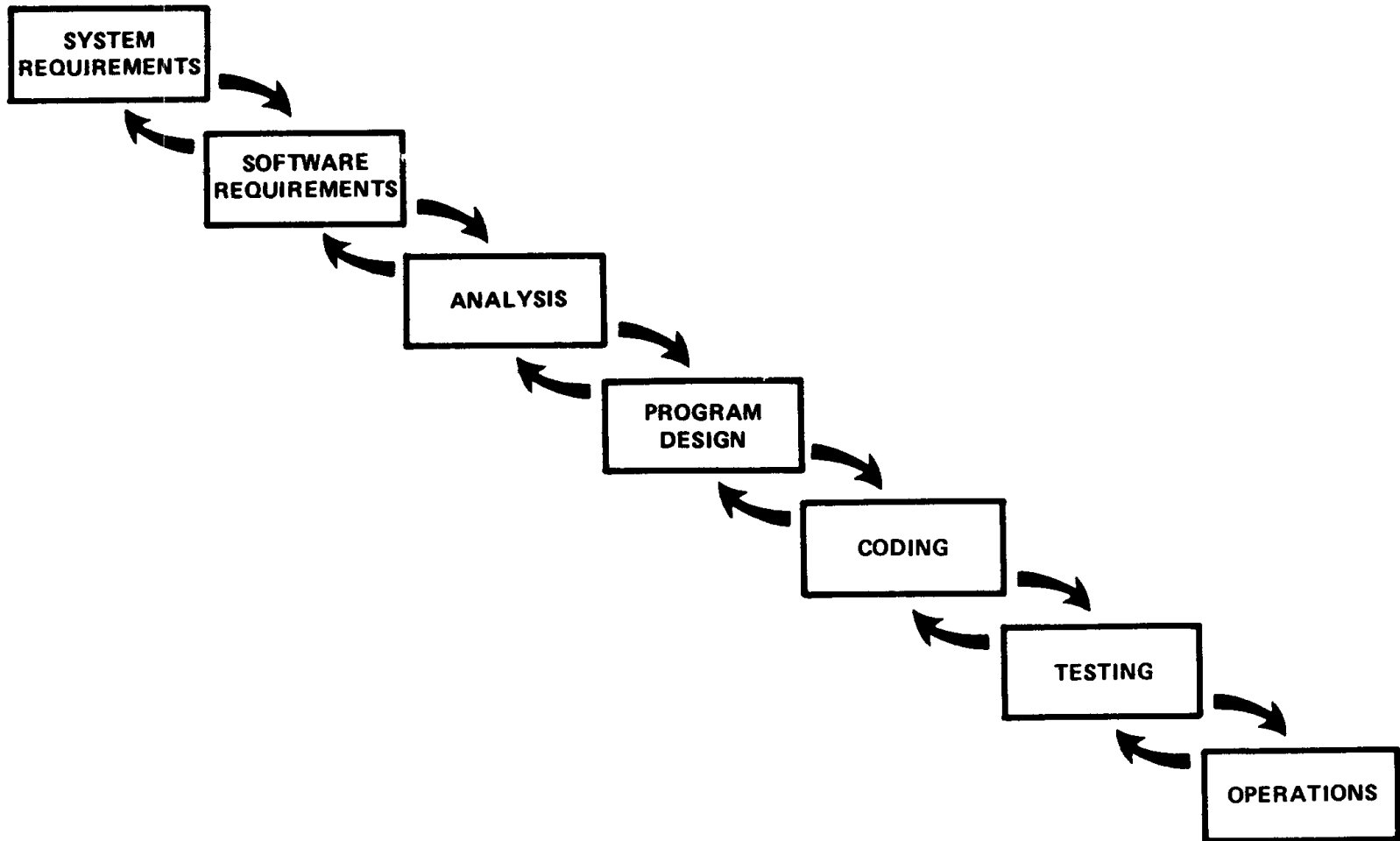
Winston. W. Royce, "Managing the development of large software systems: concepts and techniques", Proceedings of the 9th International Conference of Software Engineering, 1987, p. 328 - 338.

Η πρώτη έκδοση του άρθρου (1970): www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf

Χαρακτηριστικά

- Γραμμική ακολουθία φάσεων, ακολουθία σαφώς καθορισμένων βημάτων
- Κάθε βήμα καταλήγει στην δημιουργία προϊόντος (έγγραφο ή κώδικας)
- Κάθε προϊόν αποτελεί τη βάση για το επόμενο βήμα
- Η ορθότητα κάθε προϊόντος μπορεί να ελεγχθεί.

Μοντέλο καταρράκτη με επανάληψη



Πλεονεκτήματα

- Διαχωρισμός του έργου σε απλούστερες φάσεις
- Κάθε φάση παράγει ένα σαφώς καθορισμένο παραδοτέο

Μειονεκτήματα

- Στην πράξη οι φάσεις αλληλεπικαλύπτονται.
- Οι χρήστες συμμετέχουν μόνο στην αρχή (απαιτήσεις) και στο τέλος (λειτουργία).
- Δεν προσαρμόζεται στις αλλαγές των απαιτήσεων: στην πράξη αλλαγές σε κάποιο στάδιο επιβάλλουν την οπισθοχώρηση και πραγματοποίηση αλλαγών σε πολλά από τα προηγούμενα στάδια
- Η σχεδίαση πρέπει να ολοκληρωθεί πριν την έναρξη της υλοποίησης (στο απλό).
- Δεν επαρκεί να επικοινωνεί το κάθε βήμα με τα γειτονικά του μόνο (στο επαναληπτικό).

The required design changes are likely to be so disruptive that the software requirements upon which the design is based and which provides the rationale for everything are violated. Either the requirements must be modified, or a substantial change in the design is required. In effect the development process has returned to the origin and one can expect up to a 100-percent overrun in schedule and/or costs.

Winston W. Royce

Structured System Analysis Design Method

Η μεθοδολογία SSADM

- Είναι μια προσέγγιση στην Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων.
- Δημιουργήθηκε από ένα Βρετανικό Γραφείο που ήταν υπεύθυνο για τη χρήση της τεχνολογίας στη Βρετανική Κυβέρνηση από το 1980

Τεχνικές (1/3)

Λογικό Μοντέλο Δεδομένων

- Τεκμηρίωση των Δεδομένων του συστήματος μέσω δημιουργίας Μοντέλου Οντοτήτων Συσχετίσεων (Entity – Relationship)

Τεχνικές (2/3)

Ροή Δεδομένων μέσω των Διαδικασιών

- Μοντελοποίηση του πως κινούνται τα δεδομένα μέσω των διαδικασιών του συστήματος, με χρήση Διαγραμμάτων Ροών Δεδομένων

Τεχνικές (3/3)

Μεταβολή στο Χρόνο

- Μοντελοποίηση των γεγονότων που επηρεάζουν κάθε οντότητα
- Μοντελοποίηση της διαδικασίας που ενεργοποιείται από κάθε γεγονός

Ιεραρχική Δομή

- Φάσεις (ανάλυση, σχεδιασμός, προαιρετικά η μελέτη σκοπιμότητας)
- Στάδια (stages)
- Βήματα (steps) σειρά εξαρτώμενων βημάτων
- Εργασίες (tasks)

Φάσεις και Στάδια (1/3)

Φάση 1: Μελέτη σκοπιμότητας

- Καθορισμός προβλήματος
- Προσδιορισμός έργου

Φάσεις και Στάδια (2/3)

Φάση 2: Ανάλυση συστήματος

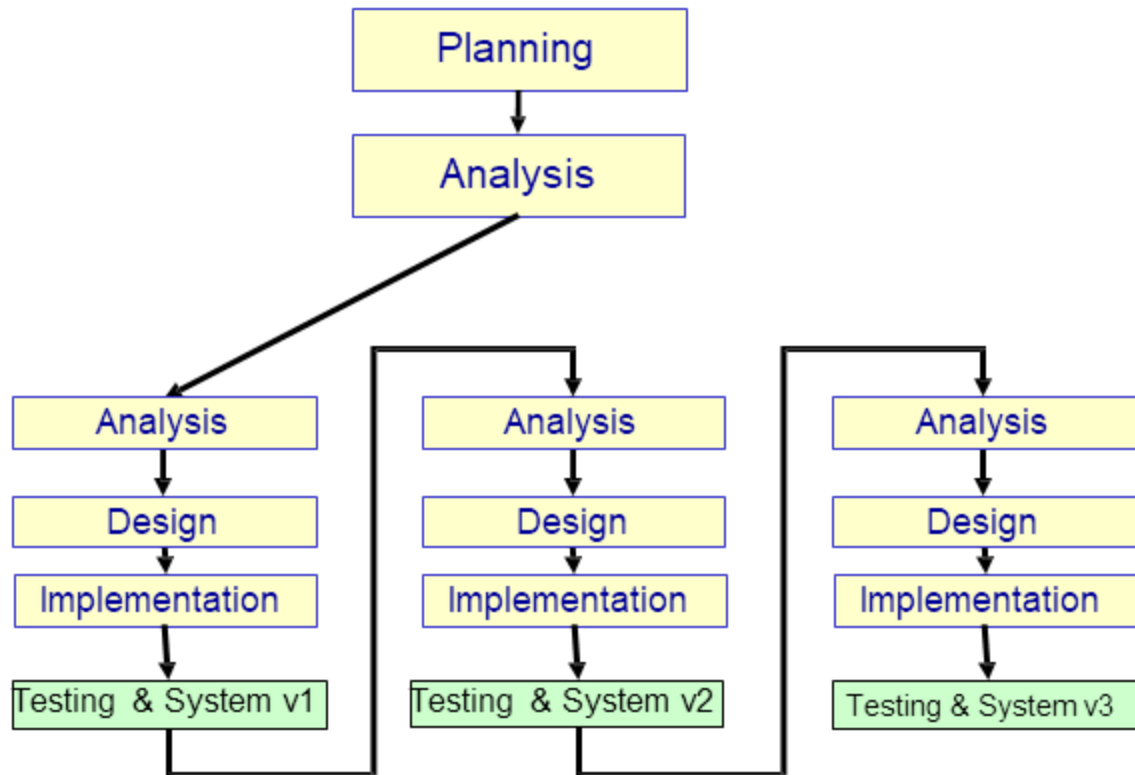
- Ανάλυση λειτουργιών και προβλημάτων συστήματος
- Προδιαγραφές απαιτήσεων
- Τεχνικές επιλογές

Φάσεις και Στάδια (3/3)

Φάση 3: Σχεδιασμός συστήματος

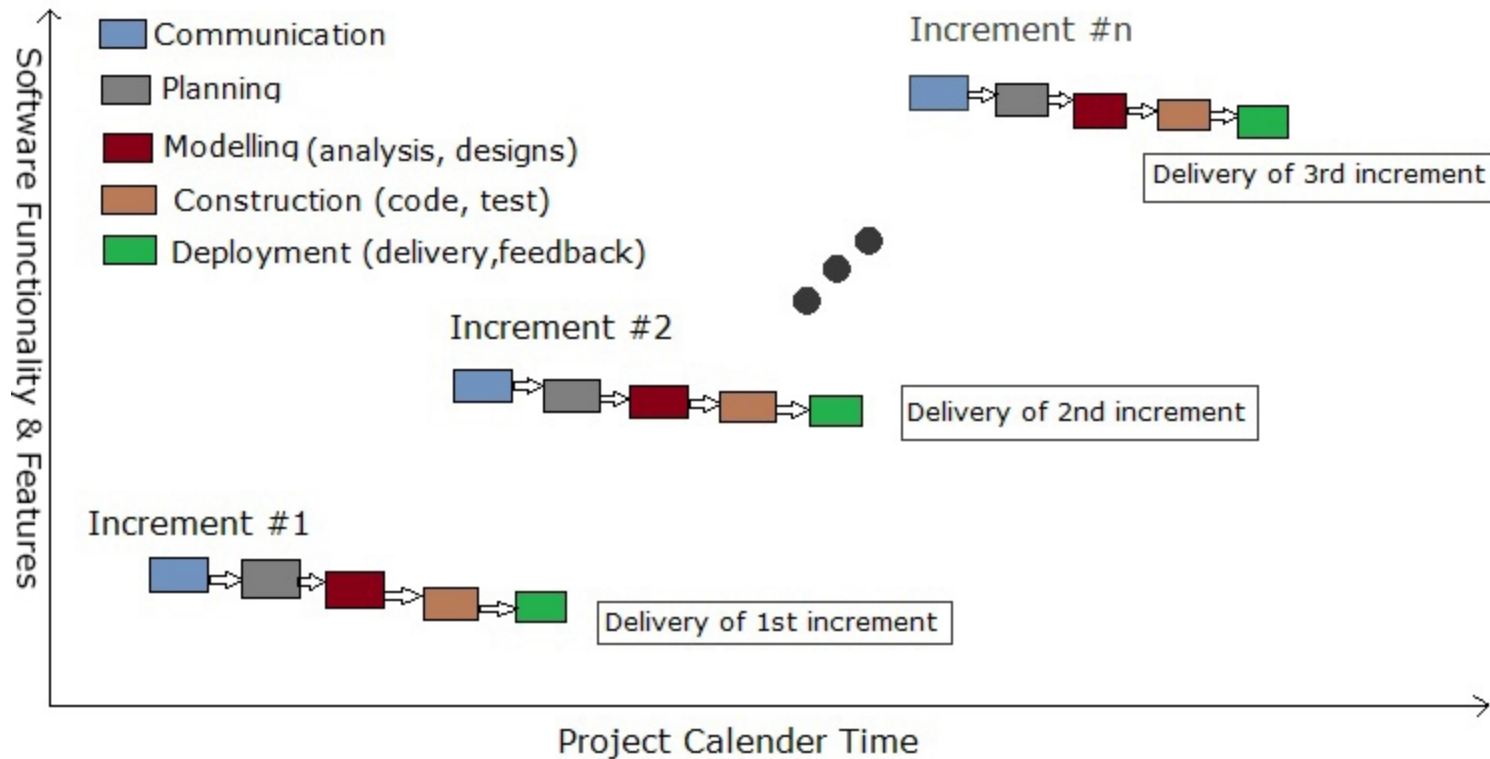
- Σχεδιασμός δεδομένων
- Σχεδιασμός διαδικασιών
- Φυσικός σχεδιασμός

Πολυφασική ανάπτυξη



Αυξητικό μοντέλο

Για σταθερές προδιαγραφές



Με δεδομένες προδιαγραφές

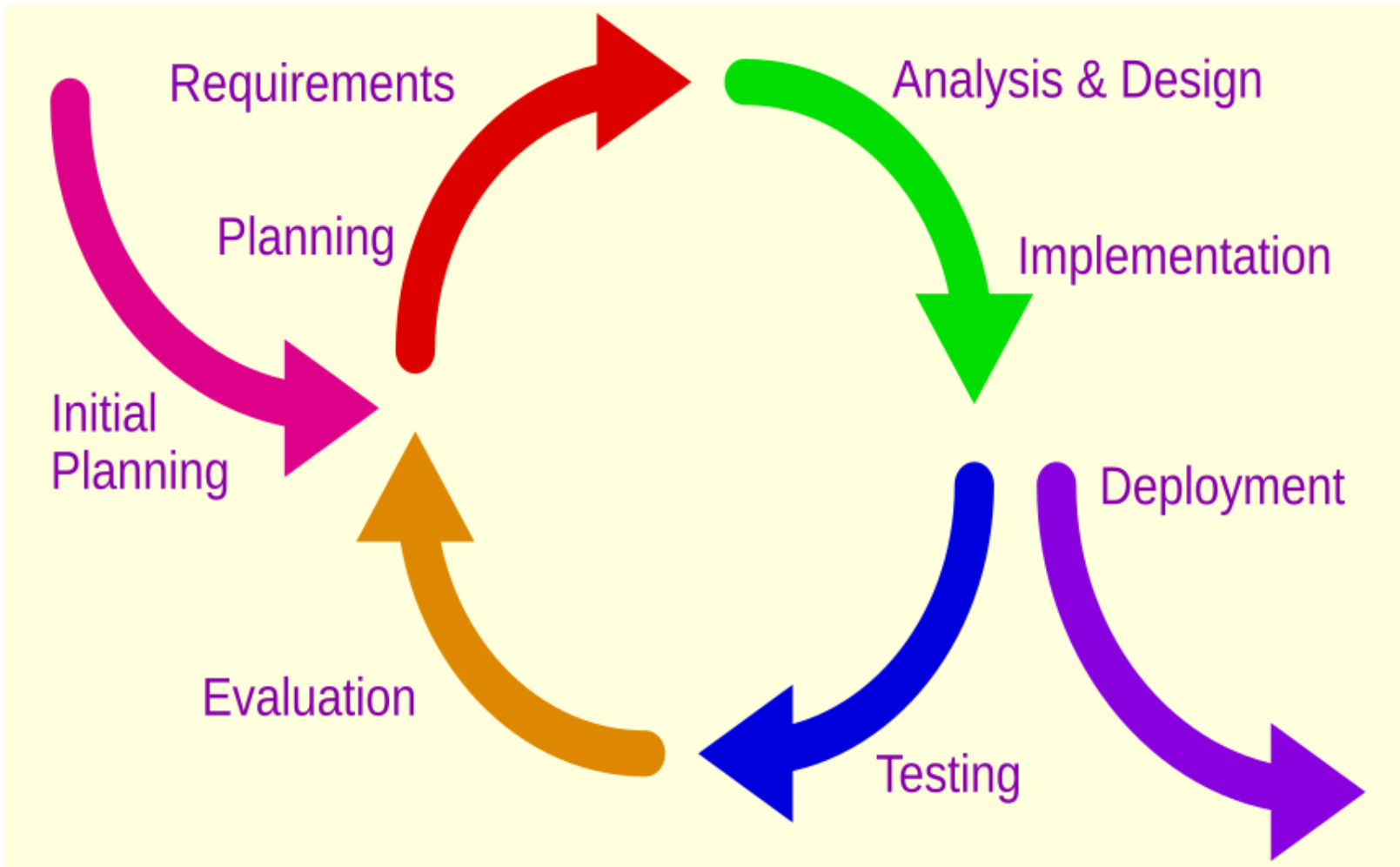
- Επιμερισμός της συνολικής εργασίας σε μικρότερες
- Κάθε έκδοση επαυξάνει την προηγούμενη (με νέες λειτουργίες και χαρακτηριστικά)
- Κάθε έκδοση είναι λειτουργική
- Μικρή ευελιξία στις αλλαγές των απαιτήσεων

Παράδειγμα

Με αυτόν τον τρόπο γίνεται η "διάθεση" (release) του λογισμικού

- "Πάγωμα" χαρακτηριστικών (feature freeze)
- Μετάβαση από τις εκδόσεις ελέγχου (π.χ. ALPHA, BETA, RELEASE CANDIDATE) στην τελική

Επαναληπτικό/εξελικτικό μοντέλο



Χαρακτηριστικά

- Επαναλαμβανόμενοι κύκλοι (iterations) σε μικρά χρονικά διαστήματα
- Με συχνές διαδράσεις με το χρήστη
- Παρόμοια με προηγούμενως, έχουμε πολλές εκδόσεις
- Οι οποίες, όμως, υλοποιούν νέες απαιτήσεις καθώς προκύπτουν/εξελισσονται

Διαδικασία πρωτοτυποποίησης

(Prototyping, Rapid Application Development)

Είναι μια επαναληπτική διαδικασία

- 1 Ανάλυση απαιτήσεων
- 2 Ανάπτυξη πρωτοτύπου (prototype)
- 3 Αποτίμηση
- 4 Βελτίωση
- 5 Επανάληψη

Πρωτότυπα (prototypes)

- Ενδείκνυται η ανάπτυξή τους σε εφαρμογές που έχουν χρηστικές διεπαφές
- Δεν ενδείκνυται σε εφαρμογές με απαιτητικούς υπολογισμούς ή εργασίες δέσμης

Οριζόντια πρωτότυπα

- Έμφαση στη διεπαφή με το χρήστη, παρά στη χαμηλού επιπέδου λειτουργικότητα
- Συνήθως UI prototypes (wireframes)
- Επιβεβαίωση χρηστικών απαιτήσεων
- Δοκιμαστικές εκδόσεις για την προσέγγιση χρηστών (πελατών ή επενδυτών, π.χ. MVP: minimum viable product)
- Βοηθούν στην εκτίμηση κόστους / χρόνου

Κάθετα πρωτότυπα

- Απόκτηση λεπτομερών απαιτήσεων για μια συγκεκριμένη λειτουργία
- Συγκέντρωση πληροφοριών για τη συμπεριφορά του συστήματος σε συγκεκριμένο περιβάλλον (π.χ. όγκος δεδομένων, ταχύτητα δικτύου, μετρήσεις απόδοσης, κλπ.)
- Αποκρυστάλλωση των απαιτήσεων για σύνθετες λειτουργίες

Rapid (throw-away) prototyping

- Σκιαγράφηση απαιτήσεων
- Σχεδιασμός πρωτοτύπου
- Χρηστική εμπειρία του, νέες απαιτήσεις
- Επανάληψη, αν χρειάζεται
- Καθορισμός των τελικών απαιτήσεων (τέλος ζωής πρωτοτύπου)

Evolutionary prototyping

- Χτίσιμο ενός στιβαρού πρωτοτύπου, που μέσω συνεχώς επεκτάσεων και βελτιώσεων, θα αποτελέσει το τελικό σύστημα
- Υλοποίηση των απόλυτα κατανοητών απαιτήσεων ανά χρονική στιγμή
- Αποτίμηση από χρήστες
- Παράλληλη ανάπτυξη των επιμέρους συστατικών του συστήματος με την ίδια λογική

Πλεονεκτήματα πρωτοτυποποίησης

- Μείωση χρόνου και κόστους
- Ανάμειξη των χρηστών

Πιθανά προβλήματα

- Ελλιπής ανάλυση
- Σύγχυση χρηστών
- Σύγχυση προγραμματιστών

Να είναι σαφής εξαρχής ο ρόλος του πρωτοτύπου στο έργο

- Προς όλους: χρήστες και προγραμματιστές
- Throw-away vs. evolutionary

Ακραία πρωτοτυποποίηση

Χρησιμοποιείται συχνά για την ανάπτυξη διαδικτυακών εφαρμογών

1. Παραγωγή στατικών σχεδίων/σελίδων
2. Προγραμματισμός λειτουργικών σελίδων με χρήση "εξομοίωσης" των back-end υπηρεσιών
3. Υλοποίηση των back-end υπηρεσιών

Rational Unified Process (RUP)

Χαρακτηριστικά

- Εξελικτικό μοντέλο με ανάδραση
- Καθοδηγείται από περιπτώσεις χρήσης (use cases)
- Είναι επικεντρωμένο στην αρχιτεκτονική
- Χρησιμοποιεί την UML σαν γλώσσα μοντελοποίησης

Δομικά στοιχεία

- Ρόλοι (ποιος) - Δικαιοδοσίες και ικανότητες
- Αποτελέσματα (τι) - Ένα παραγόμενο από το έργο αποτέλεσμα
- Ενέργειες (πώς) - Μια ενότητα εργασίας που έχει ανατεθεί σε κάποιο ρόλο

6 βασικές αρχές

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

3 βοηθητικές αρχές

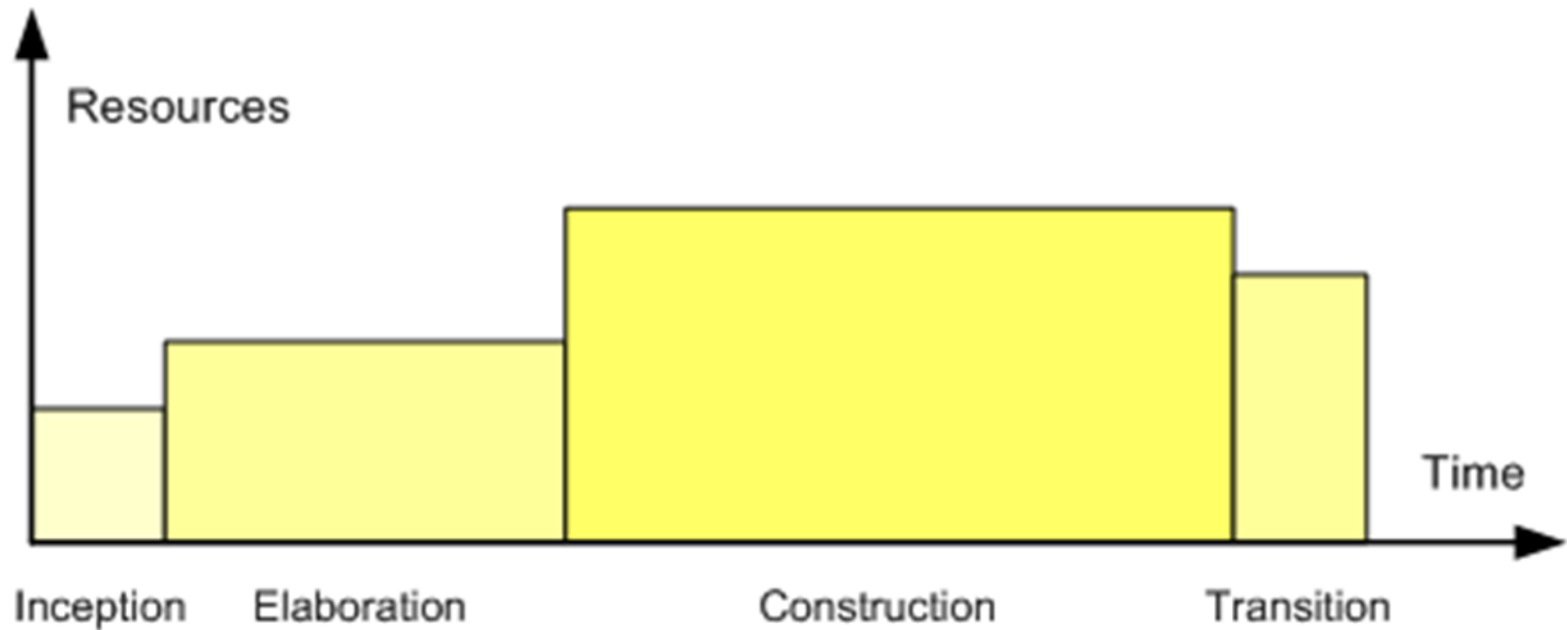
- Configuration & change management
- Project management
- Environment management and setup

4 φάσεις

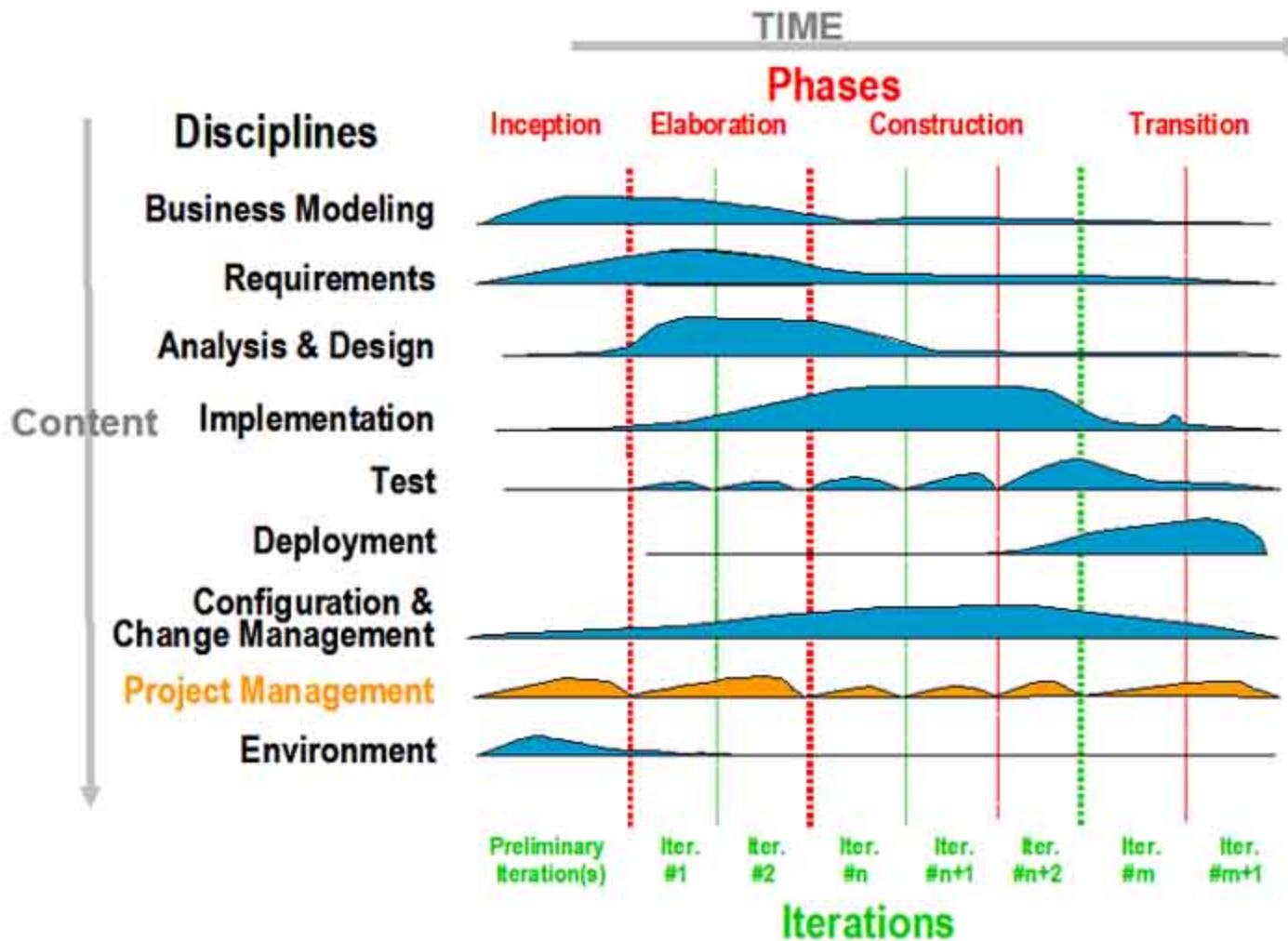
- Inception: Ορισμός και έκταση του έργου
- Elaboration: Κατάστρωση μεθόδου υλοποίησης του έργου, μοντελοποίηση χαρακτηριστικών του έργου, ορισμός της αρχιτεκτονικής του συστήματος
- Construction: Υλοποίηση του έργου
- Transition: Εγκατάσταση κι ολοκλήρωση του συστήματος στο λειτουργικό του περιβάλλον

Τα σχετικά μεγέθη των φάσεων

Σε ένα τυπικό έργο υλοποιημένο με RUP



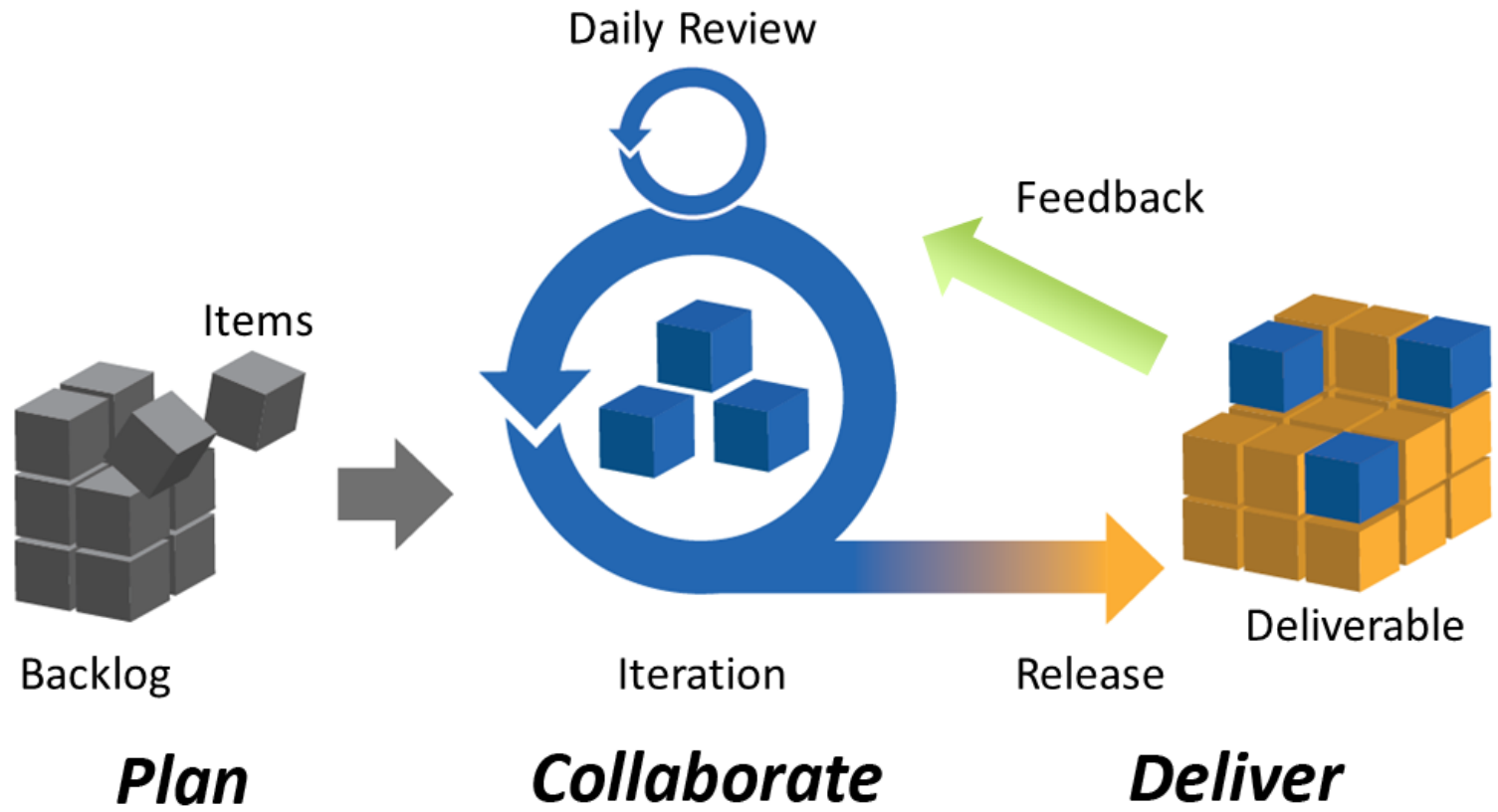
Η συνολική εικόνα



Ευέλικτο (agile) μοντέλο

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.
- Manifesto for Agile Software Development - <http://agilemanifesto.org/>



Agile Project Management: Iteration

12 Αρχές

1. Customer satisfaction by early and continuous delivery of valuable software
2. Welcome changing requirements, even in late development
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers

5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the primary measure of progress
8. Sustainable development, able to maintain a constant pace

9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Best architectures, requirements, and designs emerge from self-organizing teams
12. Regularly, the team reflects on how to become more effective, and adjusts accordingly

Scrum



Scrum

- Μικρή ομάδα
- Πολυ-λειτουργική / πολυμορφική
- Αυτο-οργάνωση
- Στον ίδιο χώρο

Αξίες

- Αφοσίωση
- Θάρρος
- Συγκέντρωση
- Ανοικτότητα - διαφάνεια
- Σεβασμός

Ρόλοι

- Product owner
 - Έμφαση στη business πλευρά
- Development team
 - Αυτο-οργάνωση (pull και όχι push)
- Scrum master
 - Όχι manager

Έννοιες και διαδικασίες

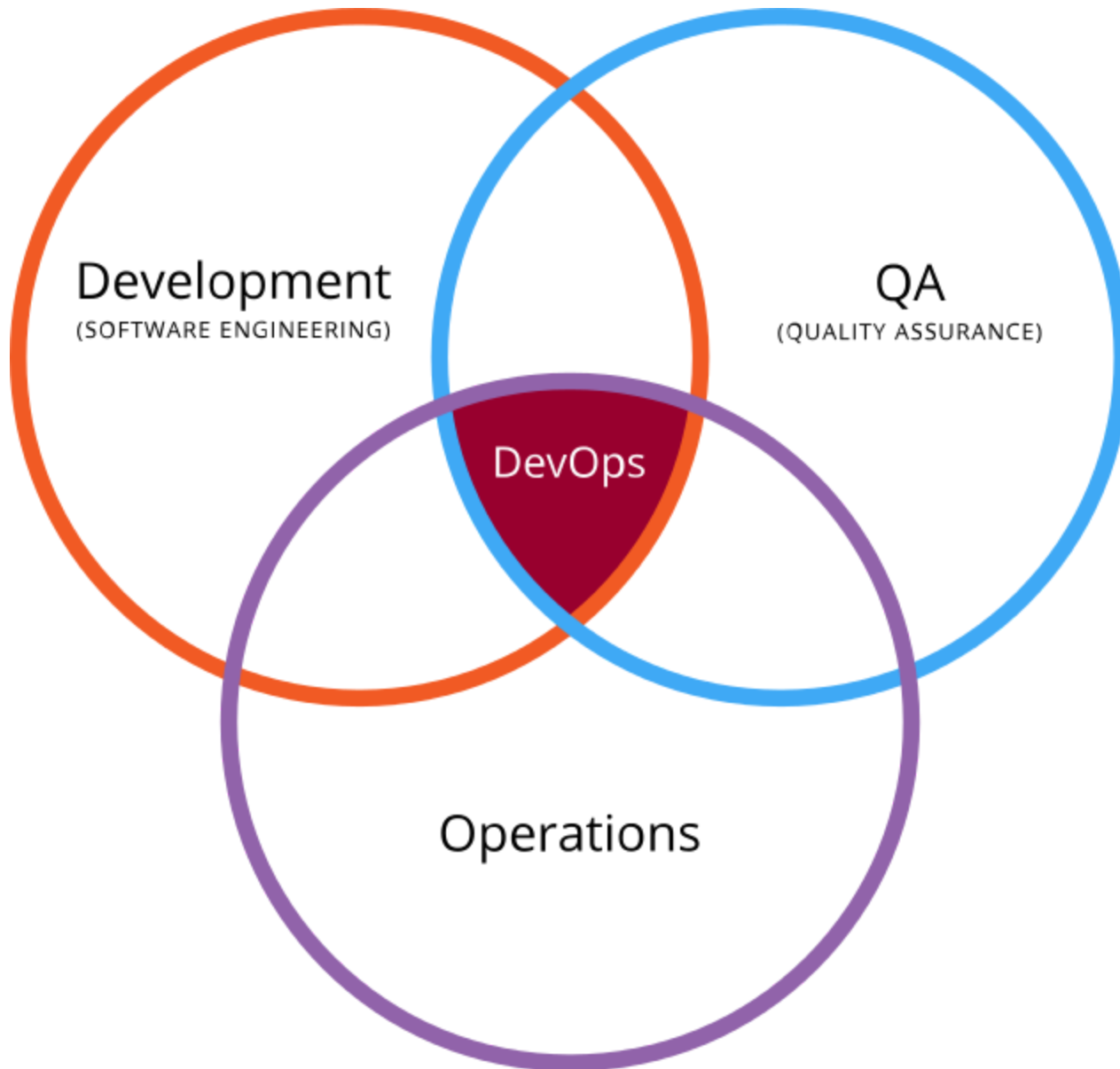
- User stories (Epics & issues)
- Product backlog
- Sprint (iteration)
- Sprint backlog

Sprint backlog

Product backlog	TODO	Doing	Done
User Story A	#1	#2	#3
Story B	#4	#6	#5
Story C			
Story D			
Story E			
Story F			

DevOps

Development + Operations



Χαρακτηριστικά

- Αυτοματισμός και παρακολούθηση των ("τελικών" κυρίως) βημάτων ανάπτυξης λογισμικού:
 - from integration, testing, releasing to deployment and infrastructure management.
- Στόχοι: μικρότεροι κύκλοι ανάπτυξης, συχνότερες εγκαταστάσεις (deployments), πιο αξιόπιστες εκδόσεις (releases), καλύτερη ευθυγράμμιση με τις επιδιώξεις του οργανισμού (business goals).

DevOps toolchain

1. Code
2. Build
3. Test
4. Package
5. Release
6. Configure
7. Monitor

