

Η Κρίση του Λογισμικού και η εξέλιξη του



Περιεχόμενα

- Τι είναι το λογισμικό;
- Ο όρος “κρίση λογισμικού”
- Η εμφάνιση του όρου “κρίση λογισμικού”
- Γιατί νοιαζόμαστε;
- Standish Group και CHAOS Report
- Υπάρχει όντως κρίση του λογισμικού;
- Οι αιτίες της κρίσης του λογισμικού
- Τρόποι εκδήλωσης κρίσης λογισμικού
- Ποια είναι η λύση;
- Μέθοδοι
- Προτεραιότητες - Standish Group
- UX - User Experience

Τι είναι το λογισμικό;

- Με τον όρο λογισμικό (software) ορίζεται η συλλογή από προγράμματα υπολογιστών, διαδικασίες και οδηγίες χρήσης που εκτελούν ορισμένες εργασίες σε ένα υπολογιστικό σύστημα.

Η Κρίση του Λογισμικού

Με τον όρο “κρίση λογισμικού” αναφερόμαστε στα προβλήματα που σχετίζονται με την ανάπτυξη λογισμικού, τα οποία οδηγούν σε προϊόντα λογισμικού εκτός χρονοδιαγράμματος, εκτός προϋπολογισμού όπου δεν καλύπτουν τις ανάγκες για τις οποίες δημιουργήθηκαν, με αποτέλεσμα να είναι δύσχρηστα και να είναι δύσκολη η επέκτασή τους ή η συντήρησή τους.

Η εμφάνιση του όρου “κρίση λογισμικού”

Στο τέλος της δεκαετία του '60 αρχίζει να γίνεται αντιληπτό από διάφορους ανθρώπους, πως δεν μπορούν να δημιουργήσουν το λογισμικό που χρειάζονται, με αποτέλεσμα την δημιουργία του όρου “κρίση λογισμικού” από ορισμένους συμμετέχοντες στην πρώτη διάσκεψη του NATO Software Engineering Conference το 1968 στο Garmisch της Γερμανίας.

Ορολογία

- Μεγάλη εταιρία – περισσότερα από 500 χιλιάδες δολάρια ανά έτος
- Μεσαία εταιρία - 200 με 500 χιλιάδες δολάρια ανά έτος
- Μικρή εταιρία - 100 με 200 χιλιάδες δολάρια ανά έτος

Γιατί νοιαζόμαστε;

Στις ΗΠΑ περισσότερα από 250 δισεκατομμύρια δολάρια ξοδεύονται για projects των 175,000 δολαρίων κάθε έτος.

Μέσο κόστος ανά εταιρία:

Μεγάλη εταιρία - \$2,322,000

Μεσαία εταιρία - \$1,331,000

Μικρή εταιρία - \$434,000

Γιατί νοιαζόμαστε;

Εκτμάται ότι οι Αμερικανικές εταιρείες και κυβερνητικές υπηρεσίες ξοδεύουν:

- 81 δισεκατομμύρια δολάρια για ακυρωμένα projects λογισμικού
- 59 δισεκατομμύρια δολάρια για προγράμματα λογισμικού που υπερβαίνουν τις αρχικές εκτιμήσεις χρόνου τους

Γιατί νοιαζόμαστε;

Επίσης βρέθηκε πως:

- Ένα εντυπωσιακό 31% των projects ακυρώνονται πριν ολοκληρωθούν

Στην επιτυχημένη πλευρά του μέσου όρου:

- 16% για projects που είναι μέσα στο χρόνο και το κόστος
- 57% των projects θα κοστίσουν το 189% των αρχικών εκτιμήσεων ο κίνδυνος μπορεί να είναι υψηλότερος για τα νεότερα έργα

Παράδειγμα

- Πόλη του Ντένβερ – απέτυχαν να παράξουν αξιόπιστο λογισμικό για τη διαχείριση των αποσκευών στο νέο αεροδρόμιο του Ντένβερ και κόστισε την πόλη 1 εκατομμύριο δολάρια το μήνα
- Αρχικά προγραμματίστηκε για τις 31 Οκτωβρίου 1993 και έληξε στις 28 Φεβρουαρίου του 1995

Standish Group και CHAOS Report

Standish Group είναι μια ανεξάρτητη διεθνής συμβουλευτική εταιρεία στον τομέα της πληροφορικής που ιδρύθηκε το 1985, γνωστή από τις εκθέσεις τους σχετικά με έργα υλοποίησης συστημάτων πληροφορικής στον δημόσιο και τον ιδιωτικό τομέα. Η εταιρεία επικεντρώνεται σε εφαρμογές λογισμικού κρίσιμης σημασίας, ειδικά με επίκεντρο τις αποτυχίες και τις πιθανές βελτιώσεις στα έργα πληροφορικής.

Standish Group και CHAOS Report

- CHAOS report είναι μια μελέτη που βασίζεται στο Πρόγραμμα CHAOS report της Standish Group σχετικά με τα ποσοστά επιτυχίας των έργων πληροφορικής και τις βέλτιστες πρακτικές διαχείρισης έργων. Η μελέτη αναφέρεται συχνά ως “the de facto authority on success rates of IT projects”. Ξεκίνησε το 1994 και ανέφερε ότι μόνο ένα εκπληκτικά χαμηλό 16,2% των έργων πληροφορικής είναι επιτυχημένο. Τα CHAOS report ταξινομούν τα έργα σε:

Τύποι Ανάλυσης

- **Successful:**

Το project ολοκληρώθηκε στην ώρα του και εντός προϋπολογισμού, με όλα τα προσδοκώμενα χαρακτηριστικά και λειτουργίες

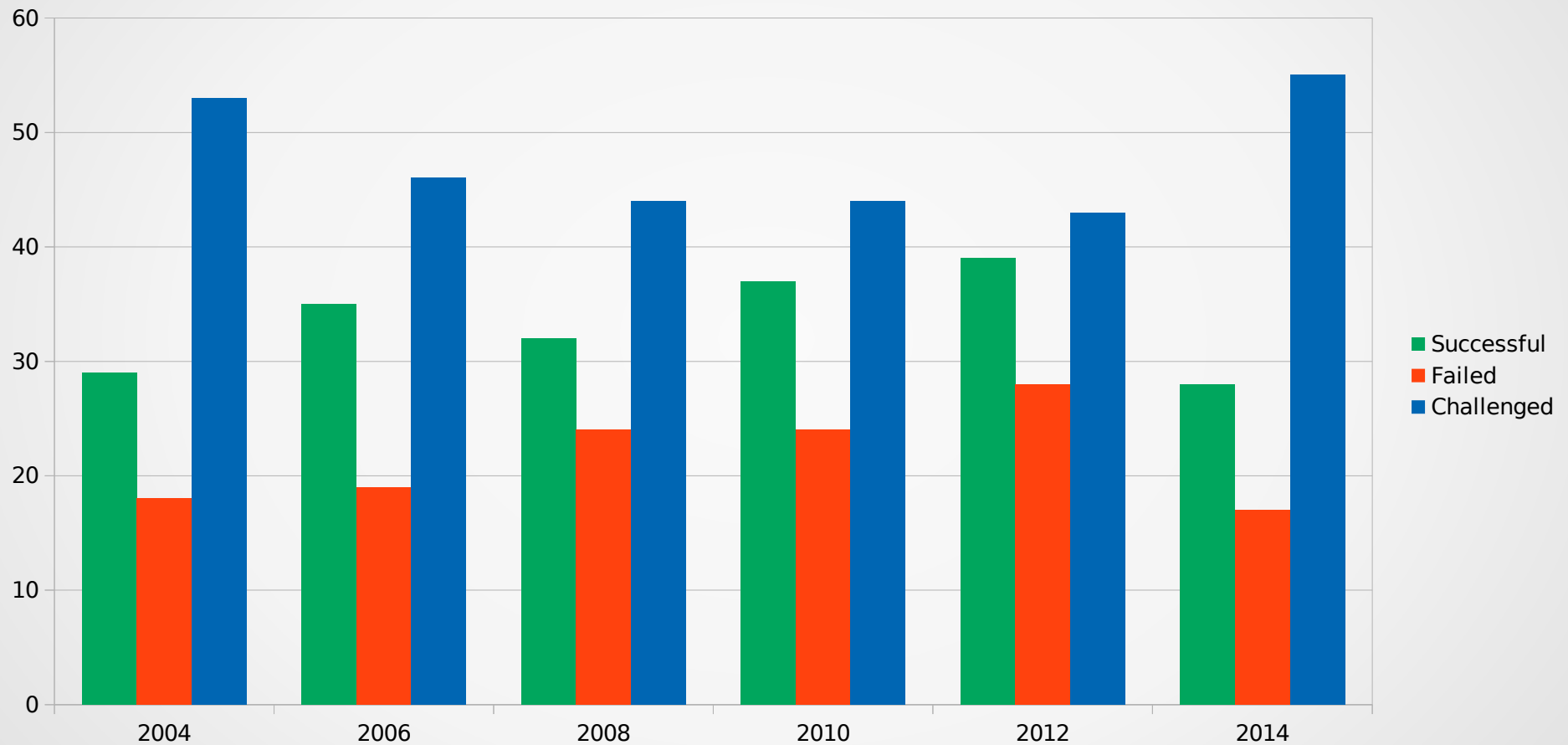
- **Challenged:**

Το project ολοκληρώθηκε αργά, ξεπέρασε τον προϋπολογισμό και δεν διέπεται από ορισμένα προσδοκώμενα χαρακτηριστικά και λειτουργίες

- **Failed:**

Ακυρώθηκε πριν την ολοκλήρωση του ή παραδόθηκε αλλά έμεινε ανεκμετάλλευτο

Στατιστικά στοιχεία

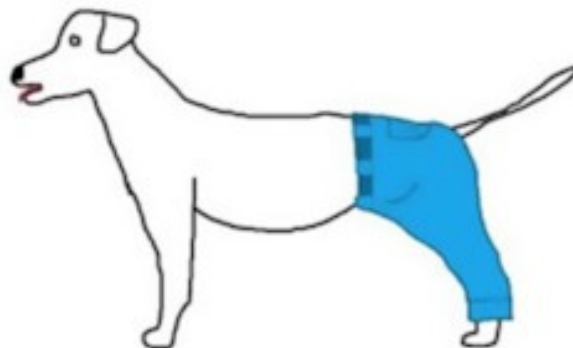
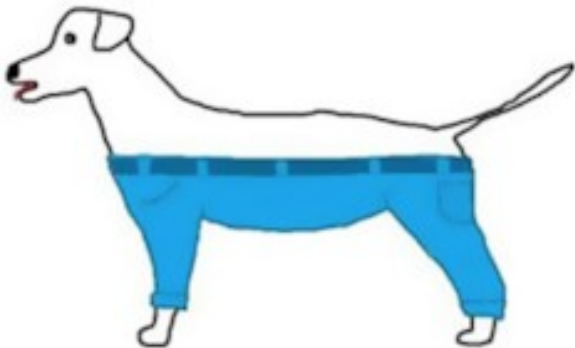


Project resolution results from CHAOS research for years 2004-2012

Ερωτήματα;;;

- Υπάρχει όντως κρίση του λογισμικού;
- Αντιπροσωπεύει την πραγματικότητα;

If a dog wore pants would he wear them
like this or like this?



Ναι ή Όχι

- Η απάντηση είναι αρκετά περίπλοκη έτσι ώστε να μην μπορούμε να απαντήσουμε με ένα ναι ή με ένα όχι.
- Είναι δύσκολο να βγάλουμε ένα συμπέρασμα καθώς δεν υπάρχει κάποια παρόμοια εταιρία ή μελέτη έτσι ώστε να μπορέσουμε να τις συγκρίνουμε.
- Οπότε το πρόβλημα μένει ανοιχτό.

any discussion of the validity of the findings. They raise those issues because most research studies conducted by academic and industry researchers arrive at data largely inconsistent with the Standish findings.

Let me say that again. Objective research study findings do not, in general, support those Standish conclusions.

Repeatedly, those researchers who have queried Standish have been rebuffed in their quest. It is apparent that Standish has not intended, at least in the past, to share much of anything about where the data used for the *Chaos Report* comes from. And that, of course, brings the validity of those findings into question.

But now there is a significant new thought regarding those Standish findings. One pair of researchers [3], combing carefully over that original Standish report, found a key description of where those findings came from. The report says, in Standish's own

the subject of failure stories eventually failed? Not much.

There is a dramatic case of déjà vu here. In the 1980s it was popular to support the notion of a software crisis by citing the *GAO Study*, a report by the U.S. Government Accounting Office that described a terrible failure rate among studied software projects. But in that case, after this had been going on for far too long, one alert researcher [1] reread the *GAO Study* and found that it admitted, quite openly, that it was a study of projects known to be failing at the time the data was gathered. Once this problem was identified, the *GAO Study* was quite quickly dropped as a citation to support the notion of software crisis. It is interesting that the first Standish study came along not too long afterward.

Is it true that the Standish study findings are as biased toward failure as the *GAO Study* results? The truth of the matter is,

Standish about this issue, to get to the heart of this critical matter, have been unsuccessful. Here, in this column, I would like to renew that line of inquiry. Standish, please tell us whether the data we have all been quoting for more than a decade really means what some have been saying it means. It is too important a topic to have such a high degree of uncertainty associated with it. **C**

REFERENCES

1. Blum, B.I. Some very famous statistics. *The Software Practitioner* (Mar. 1991).
2. Glass, R.L. IT failure rates—70 percent or 10–15 percent? *IEEE Software* 22, 3 (May–June 2005).
3. Jorgensen, M. and Molokken, K. How large are software cost overruns? A review of the 1994 *Chaos Report*. *Information and Software Technology* 48, 4 (Apr. 2006).
4. Standish Group International. *The Chaos Report*; www.standishgroup.com/sample_research/PDFpages/Chaos1994.pdf.

ROBERT L. GLASS (rlglass@acm.org) is the publisher/editor of *The Software Practitioner* newsletter and editor emeritus of Elsevier's *Journal of Systems and Software*. He is currently an honorary professor in the ARC Center for Complex Systems at Griffith University, Brisbane, Australia.

Συνέντευξη: Jim Johnson

Η συνέντευξη έγινε 25 Αυγούστου το 2006 από τον Deborah Hartmann Preuss

InfoQ: Do you think the CHAOS sample is representative of application development in general?

JJ: Yes

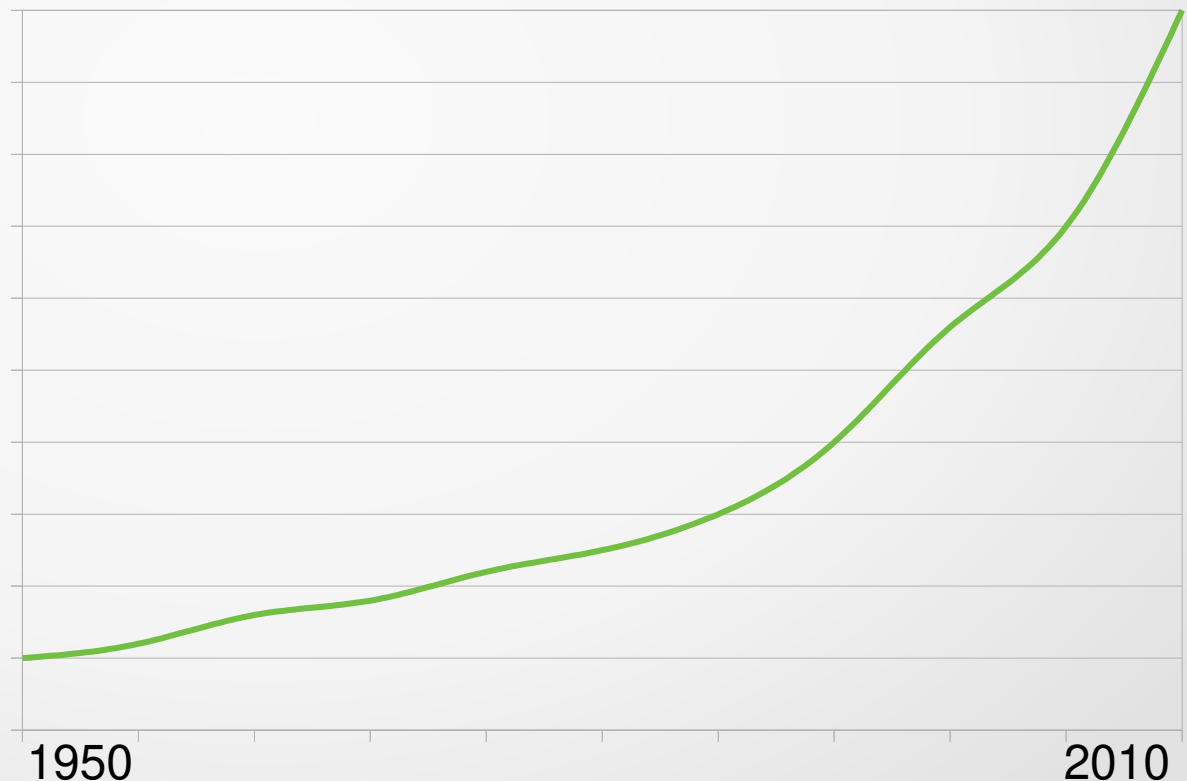
Οι αιτίες της κρίσης του λογισμικού

- Αυξανόμενη ζήτηση για λογισμικό

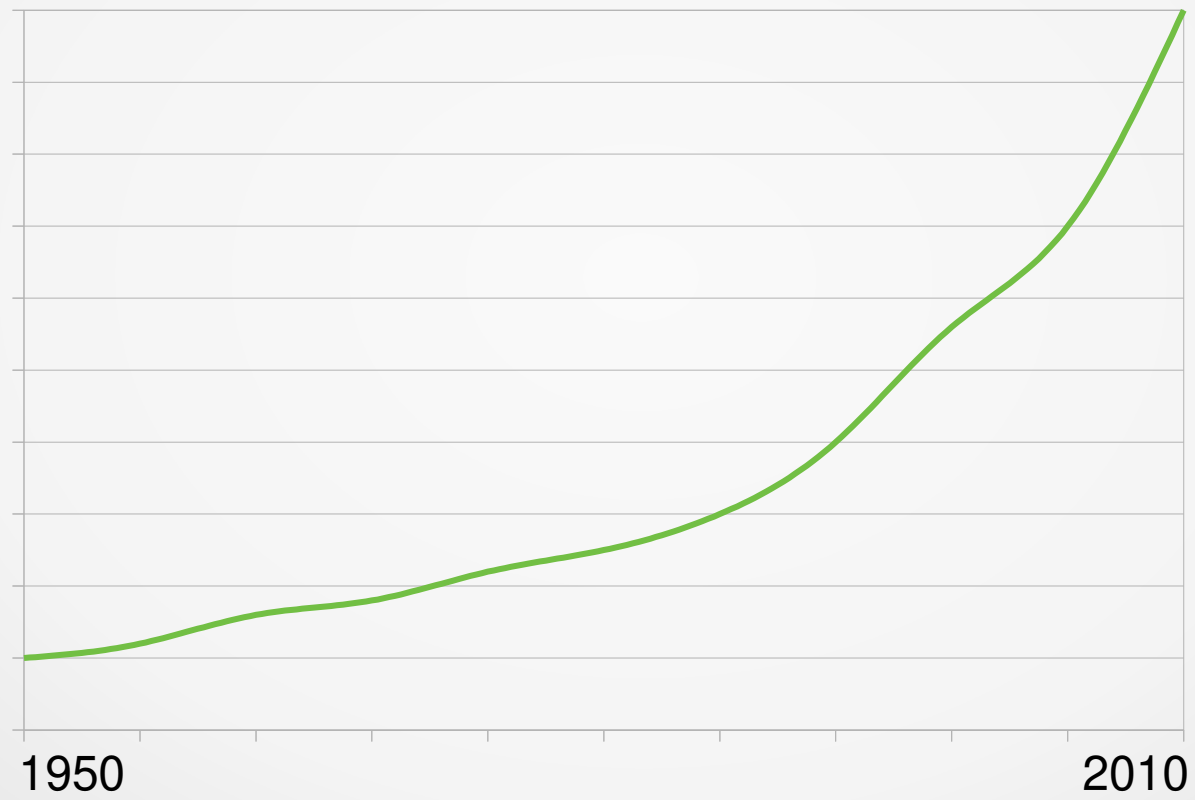
Οι σημαντικότεροι παράγοντες:

- Η αντικατάσταση του υλικού από το λογισμικό για την λύση προβλημάτων
- Η χρήση λογισμικού σε όλες τις εταιρίες και η χρήση εξειδικευμένου λογισμικού
- Η χρήση λογισμικού από τον “καθημερινό άνθρωπο”
- Και η εμφάνιση έξυπνων συσκευών.

Growth in NASA's software demand



Growth in NASA's software demand



Οι αιτίες της κρίσης του λογισμικού

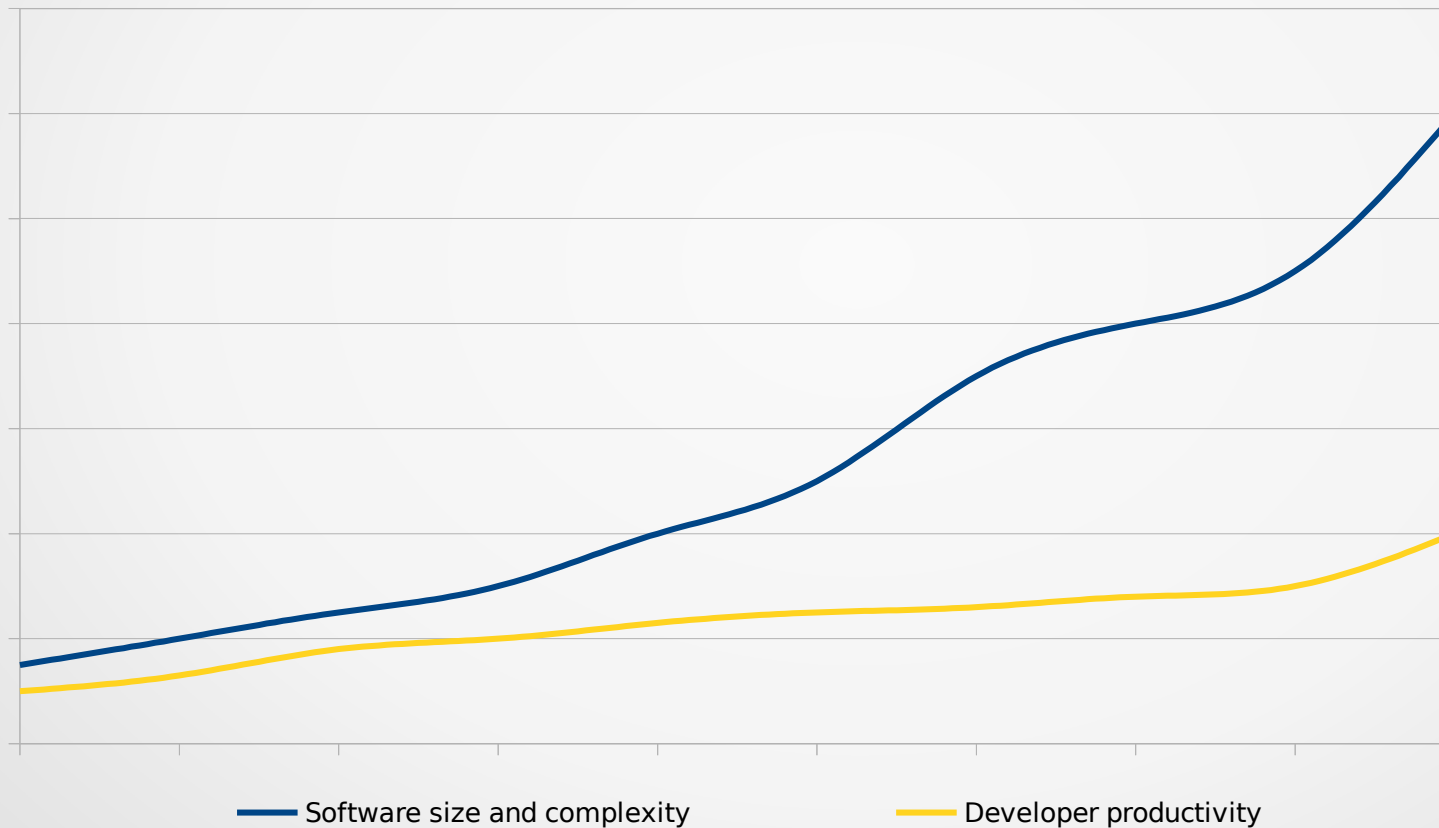
- Increase of Development effort

Με αυτόν τον όρο αναφερόμαστε στην προσπάθεια που χρειάζεται για την ανάπτυξη λογισμικού, η οποία έχει αυξηθεί λόγω της αυξημένης πολυπλοκότητας του λογισμικού.

Size	Example	
10^2 LOC	Class exercise	Programming effort
10^3 LOC	Small project	
10^4 LOC	Term project	
10^5 LOC	Word processor	Software engineering effort
10^6 LOC	Operating system	
10^7 LOC	Distributed system	

Οι αιτίες της κρίσης του λογισμικού

- Slow Developer's Productivity Growth



Οι αιτίες της κρίσης του λογισμικού

- Αδυναμία κατανόησης του πελάτη
- Δύσχρηστο γραφικό περιβάλλον
- Ακατάλληλοι μέθοδοι
- Φτωχά σχέδια
- Φτωχές διαδικασίες σχεδιασμού
- Φτωχή αρχηγεία

Τρόποι εκδήλωσης κρίσης λογισμικού

- Projects που υπερβαίνουν τον προϋπολογισμό
- Projects εκτός χρονοδιαγράμματος
- Αναποτελεσματικό λογισμικό
- Λογισμικό χαμηλής ποιότητας
- Λογισμικό που συχνά δεν πληρεί τις απαιτήσεις
- Απρόσιτα projects με κώδικας που δύσκολα διατηρείται
- Λογισμικό που δεν παραδόθηκε ποτέ

Ποια είναι η λύση;

- Δεν υπάρχει κάποια συγκεκριμένη λύση
- Υπάρχουν μέθοδοι όπου προσπαθούν να αυξήσουν τις πιθανότητες της επιτυχίας του project και προτεραιότητες

Μέθοδοι	Successful	Challenge d	Failed
Lean	72%	21%	7%
Agile	64%	30%	6%
Iterative	65%	28%	7%
Ad hoc	50%	35%	15%
Traditional	49%	32%	18%

Μέθοδοι

- Lean

Το Lean είναι μια ετικέτα που εφαρμόζεται σε μια νοοτροπία / φιλοσοφία με γνώμονα την ποιότητα που θα λάβει ο πελάτης. Μια διαδικασία Lean προσπαθεί συνεχώς να βελτιστοποιεί την ποιότητα που θα λάβει ο πελάτης, ελαχιστοποιώντας παράλληλα τα απόβλητα (τα οποία μπορεί να μετρηθούν με βάση το χρόνο, την ποιότητα και το κόστος). Τελικά, το “ταξίδι” Lean είναι η ανάπτυξη ενός εκπαιδευτικού οργανισμού. Παραδείγματα μεθόδων / διαδικασιών Lean περιλαμβάνουν τις Kanban και Scrumban.

Μέθοδοι

- Agile

Σε ένα πρόγραμμα ανάπτυξης λογισμικού Agile, η ομάδα ακολουθεί μια επαναληπτική διαδικασία που είναι επίσης ελαφριά, εξαιρετικά συνεργατική, αυτο-οργανωτική και επικεντρωμένη στην ποιότητα. Παραδείγματα μεθόδων Agile περιλαμβάνουν το Scrum, XP, και Disciplined Agile Delivery (DAD).

Μέθοδοι

- Iterative

Σε ένα επαναληπτικό πρόγραμμα ανάπτυξης λογισμικού, η ομάδα ακολουθεί μια διαδικασία που οργανώνεται σε περιόδους που συχνά αναφέρονται ως επαναλήψεις ή χρονικά πλαίσια. Σε οποιαδήποτε δεδομένη ημέρα του έργου, τα μέλη της ομάδας μπορεί να συγκεντρώνουν απαιτήσεις, να σχεδιάζουν, να γράφουν κώδικα, να δοκιμάζουν και ούτω καθεξής. Ένα παράδειγμα μιας επαναληπτικής διαδικασίας είναι η RUP. Τα ευκίνητα έργα, τα οποία ορίζονται ως επαναληπτικά έργα που εκτελούνται με εξαιρετικά συνεργατικό και ελαφρύ τρόπο, αντιμετωπίζονται με καθυστέρηση.

Μέθοδοι

- Ad hoc

Σε ένα έργο ανάπτυξης λογισμικού ad hoc, η ομάδα δεν ακολουθεί μια καθορισμένη διαδικασία.

- Traditional

Σε ένα παραδοσιακό πρόγραμμα ανάπτυξης λογισμικού, η ομάδα ακολουθεί μια σταδιακή διαδικασία, όπου προσδιορίζονται πρώτα οι απαιτήσεις, τότε ορίζεται η αρχιτεκτονική / σχεδιασμός, στη συνέχεια γίνεται η κωδικοποίηση, κατόπιν δοκιμάζεται και αναπτύσσεται. Οι παραδοσιακές διαδικασίες συχνά αναφέρονται ως "καταρράκτες" ή απλά "σειριακές" διαδικασίες.

Προτεραιότητες - Standish Group

1. User Involvement
2. Executive Management Support
3. Clear Business Objectives
4. Optimizing Scope
5. Agile Process
6. Project Manager Expertise
7. Financial Management
8. Skilled Resources
9. Formal Methodology
10. Standard Tools and Infrastructure

UX - User Experience

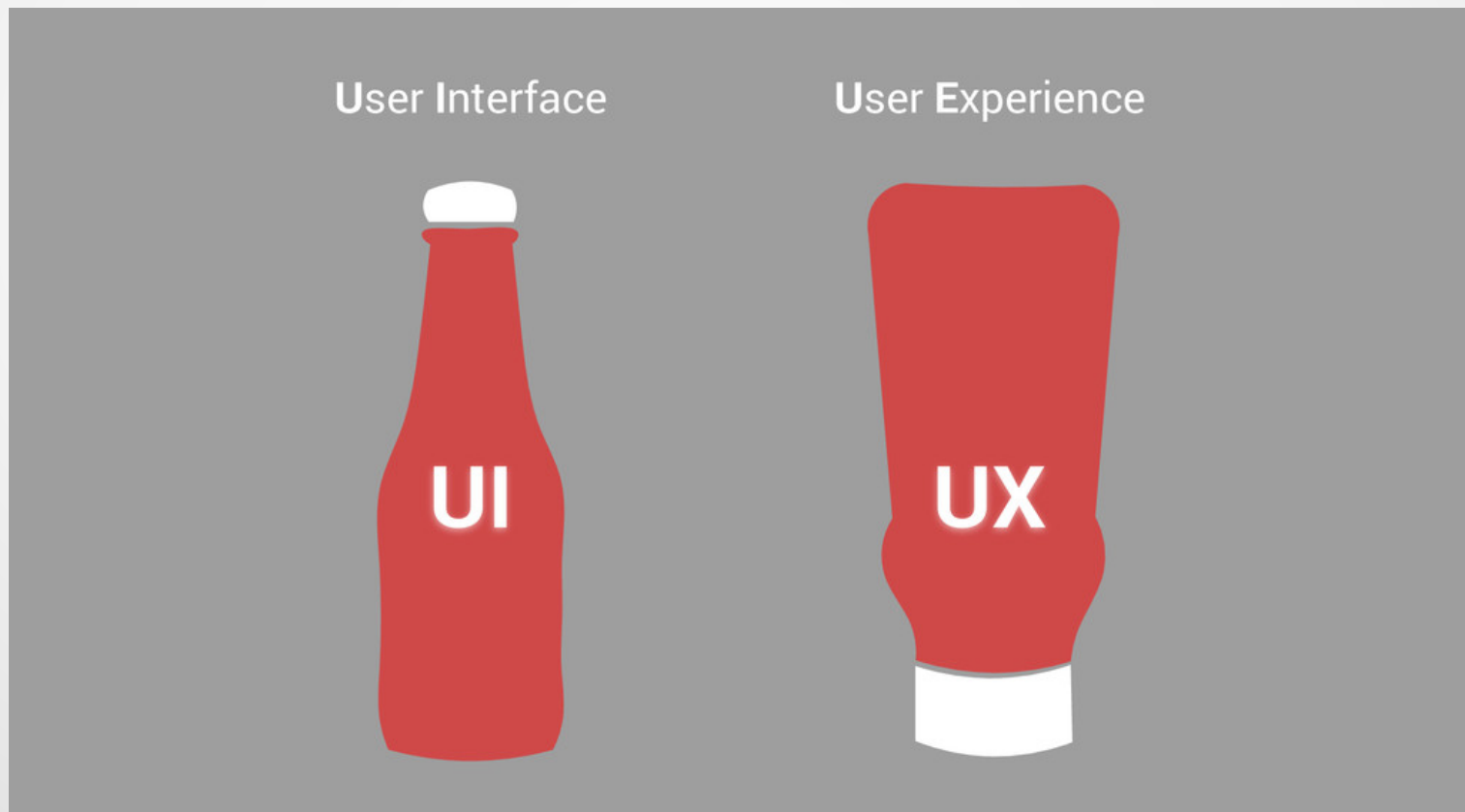
Ο σχεδιασμός της εμπειρίας χρήστη (UX, UXD, UED ή XD) είναι η διαδικασία βελτίωσης της ικανοποίησης των χρηστών από ένα προϊόν, βελτιώνοντας τη χρηστικότητα, την προσβασιμότητα και την ευχαρίστηση που παρέχονται στην αλληλεπίδραση με το προϊόν. Ο σχεδιασμός εμπειρίας χρήστη περιλαμβάνει τον παραδοσιακό σχεδιασμό αλληλεπίδρασης μεταξύ ανθρώπων και ηλεκτρονικών υπολογιστών (HCI), επεκτείνεται με την αντιμετώπιση όλων των πτυχών ενός προϊόντος ή μιας υπηρεσίας όπως γίνεται αντιληπτή από τους χρήστες.

UX και Ελλάδα

- Στην Ελλάδα άρχισε να εμφανίζεται αυτός ο όρος κατά το 2014 και σήμερα υπάρχουν κάποιες εταιρείες / οργανισμοί όπου ασχολούνται με αυτόν τον τομέα.
- Ένας από αυτούς είναι ο theUXProdigy που δημιουργήθηκε από τον Παναγιώτη Ζαχαρία.

Γιατί είναι σημαντικό το UX;

Είναι σημαντικό γιατί καθορίζει την άποψη του χρήστη και αυτό μπορεί να έχει μεγάλη επίπτωση στο προϊόν ή στην εταιρία.



Πηγές

- Software: <https://en.wikipedia.org/wiki/Software>
 - Software Crisis: https://en.wikipedia.org/wiki/Software_crisis
<https://getlevelten.com/wiki/chaos-report>
<https://www.youtube.com/watch?v=0b5vp4Z2PKE>
<https://www.youtube.com/watch?v=Ao-Ws0C74R4>
<https://www.youtube.com/watch?v=kJP9qfNO7Rs>
<http://www.wisegeek.com/what-is-a-software-crisis.htm#didyouknowout>
<http://ecomputernotes.com/software-engineering/software-crisis>
http://tomandmaria.com/Tom/Writing/SoftwareCrisis_SofiaDRAFT.pdf
<http://www.studymode.com/essays/Software-Crisis-429194.html>
<https://www.informit.com/articles/article.aspx?p=2246403>
<http://slideplayer.com/slide/3936168/>
- Paper: Productivity Impacts of Software Complexity and Developer Experience by Geoffrey K. Gill and Chris F. Kemerer (MIT Sloan School WP #3107-90 January 1990)

Πηγές

- Standish Group και CHAOS Report:

https://en.wikipedia.org/wiki/Standish_Group

<http://www.wordyard.com/2006/08/02/standishs-chaos-report-and-the-software-crisis/>

<https://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS>

https://www.researchgate.net/publication/220422763_The_Standish_Report_Does_It_Really_Describe_a_Software_Crisis

<http://www.drdoobbs.com/architecture-and-design/the-non-existent-software-crisis-debunk/240165910>

<https://cacm.acm.org/magazines/2006/8/5855-the-standish-report/abstract>

<https://www.infoq.com/articles/standish-chaos-2015>

There is no software engineering crisis by Alex Colburn, Jonathan Hsieh, Matthew Kehrt and Aaron Kimball (January 16, 2008)

The Standish Report: Does It Really Describe a Software Crisis? By Robert L. Glass

Πηγές

- UX - User Experience:

https://en.wikipedia.org/wiki/User_experience_design

<https://www.theuxprodigy.com/>

<https://www.youtube.com/watch?v=qtCEoGyfsxk>