



eXtreme Programming

Gianluca Demartini

26 aprile 2004

Scaletta

- ◆ XP: definizione
- ◆ XP: obiettivo
- ◆ XP: i 4 valori
- ◆ XP: le 12 pratiche
- ◆ XP: l'ambiente di lavoro
- ◆ XP: i ruoli principali
- ◆ XP: i problemi
- ◆ XP: opinioni personali

Gianluca Demartini

2

XP: definizione

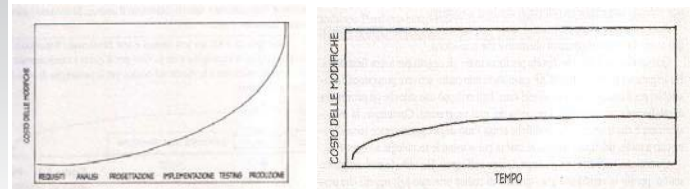
- ◆ L'Extreme Programming (o XP) è un insieme di valori, principi e pratiche per un rapido sviluppo di software di alta qualità, che fornisca il più alto valore possibile per i clienti nel più veloce modo possibile (www.jera.com)
- ◆ L'XP è un modo leggero, efficace, a basso rischio, flessibile, prevedibile, scientifico e divertente per sviluppare il software (Erich Gamma)

Gianluca Demartini

3

XP: obiettivo

- ◆ Risolvere i principali problemi del classico modo di sviluppare software
- ◆ Abbassare la curva del costo delle modifiche nel tempo



Gianluca Demartini

4

Scaletta

- ◆ XP: definizione
- ◆ XP: obiettivo
- ◆ XP: i 4 valori
- ◆ XP: le 12 pratiche
- ◆ XP: l'ambiente di lavoro
- ◆ XP: i ruoli principali
- ◆ XP: i problemi
- ◆ XP: opinioni personali

Gianluca Demartini

5

XP: i 4 valori (1/2)

- ◆ **Comunicazione:**
 - ◆ i problemi possono essere causati da qualcuno che non ha riferito qualcosa di importante a qualcun altro
 - ◆ programmatori, manager e cliente devono comunicare: serve un "coach"
- ◆ **Semplicità:**
 - ◆ bisogna fare la cosa più facile che possa funzionare
 - ◆ il codice sorgente rivela le intenzioni che ci sono dietro a chiunque lo legga

Gianluca Demartini

6

XP: i 4 valori (2/2)

- ◆ **Feedback:**
 - ◆ serve a non far perdere l'ottimismo ai programmatori
 - ◆ i clienti sono al corrente dello stato attuale del sistema
 - ◆ serve a comunicare meglio!
- ◆ **Coraggio:**
 - ◆ si può buttare via codice
 - ◆ si può provare a ridurre la complessità dell'intero sistema
 - ◆ funziona solo insieme agli altri valori!

Gianluca Demartini

7

Scaletta

- ◆ XP: definizione
- ◆ XP: obiettivo
- ◆ XP: i 4 valori
- ◆ XP: le 12 pratiche
- ◆ XP: l'ambiente di lavoro
- ◆ XP: i ruoli principali
- ◆ XP: i problemi
- ◆ XP: opinioni personali

Gianluca Demartini

8

XP: le 12 pratiche

- ◆ Planning Game
- ◆ Brevi cicli di rilascio
- ◆ Metafora
- ◆ Semplicità di progetto
- ◆ Testing
- ◆ Refactoring
- ◆ Programmazione a coppie
- ◆ Proprietà collettiva
- ◆ Integrazione continua
- ◆ Settimana di quaranta ore
- ◆ Cliente sul posto
- ◆ Standard di codifica

Gianluca Demartini

9

XP le 12 pratiche: Brevi cicli di rilascio

- ◆ Ogni ciclo di rilascio dovrebbe essere il più piccolo possibile e dovrebbe contenere le funzionalità di maggior valore
- ◆ Più o meno 1 o 2 mesi (in base al software che si sta sviluppando)
- ◆ Produrre un piccolo insieme di funzionalità utili

Gianluca Demartini

10

XP le 12 pratiche: Metafora

- ◆ Ogni progetto è guidato da una metafora che lo racchiude
- ◆ La metafora fornisce una convenzione per l'assegnazione dei nomi, in modo che siano facili da ricordare
- ◆ E' più facile parlare con il cliente in termini della metafora
- ◆ La metafora aiuta a capire gli elementi fondamentali e le loro relazioni

Gianluca Demartini

11

XP le 12 pratiche: Semplicità di progetto (1/2)

- ◆ Domani i requisiti cambieranno, quindi il progetto deve essere il più semplice tale che soddisfi quelli odierni
- ◆ Implementate quello che serve, solo quando serve
- ◆ Il progetto giusto deve: far funzionare tutti i test e non avere alcuna logica duplicata

Gianluca Demartini

12

XP le 12 pratiche: Semplicità di progetto (2/2)

- ◆ Esempio:
 - ◆ “C'è bisogno di una finestra per mostrare i testi: si può fare una finestra molto evoluta che sia in grado di adeguare le proprie dimensioni e il numero di interruzioni di linea (2 giorni di lavoro) in base al testo, oppure una più semplice che soddisfi soltanto lo scopo di mostrare un testo (20 minuti di lavoro) e migliorarla se sarà necessario”
- ◆ Bisogna implementare la finestra più semplice

Gianluca Demartini

13

XP le 12 pratiche: Planning Game (1/2)

- ◆ Bisogna determinare velocemente le funzionalità (scritte come **storia**) del prossimo rilascio del software:
 - ◆ gli sviluppatori stimano il tempo necessario per implementare le funzionalità
 - ◆ il cliente sceglie con che ordine implementare le storie
- ◆ L'obiettivo è di produrre il massimo *business value* il più velocemente possibile

Gianluca Demartini

14

XP le 12 pratiche: Planning Game (2/2)

- ◆ Esempio di storia
- ◆ Un sistema per le paghe dei dipendenti:
 - ◆ “Quando il tasso di merito per un dipendente cambia durante un periodo di paga, paghiamo la prima settimana usando il vecchio tasso e la seconda usando il nuovo. Serve un meccanismo automatico per fare ciò”
- ◆ Il programmatore che si impegna a realizzare questa storia deve stimare il tempo che impiegherà ad implementarla

Gianluca Demartini

15

XP le 12 pratiche: Testing (1/2)

- ◆ Prima che i programmatori sviluppino una funzionalità, devono scriverne i relativi test (uno per ogni metodo che possa fallire)
- ◆ Quando i test funzioneranno, la funzionalità sarà stata implementata
- ◆ Due tipi di test: i clienti scrivono i **test funzionali**, i programmatori scrivono i **test unitari**

Gianluca Demartini

16

XP le 12 pratiche: Testing (2/2)

- ◆ Due tipi di test:
 - ◆ I **test unitari** sono dei test che vanno a verificare la correttezza direttamente del codice, in ogni sua piccola parte (JUnit)
 - ◆ I **test funzionali** sono dei test che vanno a verificare che il software nel suo insieme funzioni correttamente

Gianluca Demartini

17

XP le 12 pratiche: Programmazione a coppie

- ◆ Tutto il codice deve essere scritto da due persone davanti allo stesso terminale:
 - ◆ uno implementa il metodo in corso e l'altro pensa a come semplificare il sistema, quali altri test implementare, ecc.
- ◆ Il codice viene riveduto e corretto mentre viene scritto: ci sono meno probabilità di fare errori
- ◆ Le coppie sono dinamiche: ogni programmatore sceglie il suo compagno

Gianluca Demartini

18

XP le 12 pratiche: Proprietà collettiva

- ◆ Ogni programmatore dovrebbe essere in grado di lavorare su qualsiasi parte del codice in ogni momento
- ◆ Chiunque veda l'opportunità di migliorare una qualsiasi parte del codice, deve farlo, in qualunque momento
- ◆ Tutti si prendono la responsabilità dell'intero sistema

Gianluca Demartini

19

XP le 12 pratiche: Standard di codifica

- ◆ I programmatori passano da una parte all'altra del sistema e ristrutturano l'uno il codice dell'altro di continuo
- ◆ Ognuno deve programmare con le stesse convenzioni (parentesi, indentazione, ecc.); Bisogna arrivare al punto di non saper dire chi ha scritto una certa parte di codice (Proprietà collettiva)
- ◆ E' facilitata la comunicazione e il refactoring

Gianluca Demartini

20

XP le 12 pratiche: Refactoring

- ◆ I programmatori si devono chiedere se:
 - ◆ si possa semplificare il sistema
 - ◆ ridurre la duplicazione di codice
 - ◆ comunicare con maggiore chiarezza attraverso il codice
- ◆ Bisogna ristrutturare il codice se si pensa che il compito che esso svolge possa essere fatto più semplicemente
- ◆ Si può fare senza pericolo perché ci sono i test automatici che ci dicono se il nuovo codice funziona come il vecchio

Gianluca Demartini

21

XP le 12 pratiche: Integrazione continua

- ◆ Il codice viene integrato ogni poche ore (un giorno di sviluppo al massimo) risolvendo i conflitti, fino a che tutti i test funzionano
- ◆ I test devono funzionare al 100% prima e dopo l'integrazione
- ◆ L'ideale è avere una macchina dedicata all'integrazione

Gianluca Demartini

22

XP le 12 pratiche: Settimana di quaranta ore

- ◆ Se il gruppo di lavoro non è fresco e pieno di energie, non può far funzionare le altre pratiche
- ◆ Non si può fare straordinario per due settimane di fila, altrimenti c'è un problema serio nel progetto
- ◆ Se poco prima di finire la giornata c'è un problema, è inutile cercarlo fino a notte fonda. Meglio farlo la mattina dopo freschi e riposati

Gianluca Demartini

23

XP le 12 pratiche: Cliente sul posto

- ◆ Un cliente reale (che userà il sistema) deve essere disponibile a rispondere alle domande, risolvere le divergenze, stabilire le priorità
- ◆ Il cliente può produrre valore per il progetto scrivendo i test funzionali
- ◆ Il cliente può continuare il suo lavoro anche se è fisicamente separato dai suoi colleghi

Gianluca Demartini

24

XP: le 12 pratiche

- ◆ Se non stai usando tutte le 12 pratiche, non stai facendo XP
- ◆ C'è un effetto sinergico nell'usarle tutte assieme: ognuna ha bisogno della altre per mantenersi in equilibrio
- ◆ Esempio: senza uno standard di codifica non si può parlare di proprietà collettiva e non è facile programmare a coppie

Gianluca Demartini

25

Scaletta

- ◆ XP: definizione
- ◆ XP: obiettivo
- ◆ XP: i 4 valori
- ◆ XP: le 12 pratiche
- ◆ XP: l'ambiente di lavoro
- ◆ XP: i ruoli principali
- ◆ XP: i problemi
- ◆ XP: opinioni personali

Gianluca Demartini

26

XP: l'ambiente di lavoro

- ◆ Serve uno spazio di lavoro aperto per il gruppo, con al centro un'area comune per la programmazione e piccoli spazi privati intorno
- ◆ Va messa una macchinetta del caffè, cibo, divani in un'area comune: il modo migliore per sbloccarsi mentre si programma è allontanarsi dal computer

Gianluca Demartini

27

XP: i ruoli principali

- ◆ Il programmatore:
 - ◆ il suo valore principale è la comunicazione: non basta che il codice funzioni correttamente, deve anche seguire gli standard di codifica, riorganizzare il codice, scrivere i test, ecc.
- ◆ Il cliente:
 - ◆ deve imparare a scrivere buone storie
 - ◆ deve prendere decisioni
 - ◆ deve imparare a scrivere test funzionali
- ◆ Il coach:
 - ◆ è il responsabile dell'intero processo: deve guidare il gruppo se questo devia dal regolare funzionamento

Gianluca Demartini

28

Scaletta

- ◆ XP: definizione
- ◆ XP: obiettivo
- ◆ XP: i 4 valori
- ◆ XP: le 12 pratiche
- ◆ XP: l'ambiente di lavoro
- ◆ XP: i ruoli principali
- ◆ XP: i problemi
- ◆ XP: opinioni personali

Gianluca Demartini

29

XP: i problemi (1/2)

- ◆ Molto meno documentazione sostituita dal dialogo (sia verso management che verso il cliente), quindi se un programmatore va via ci possono essere problemi
- ◆ Serve una "cultura aziendale": tutti devono considerare la fase di programmazione come essenziale (non il 20% del progetto)

Gianluca Demartini

30

XP: i problemi (2/2)

- ◆ Funziona solo con un numero non elevato di programmatori (una dozzina) con conseguente limitazione alle dimensioni del SW sviluppabile
- ◆ Serve l'ambiente fisico adatto

Gianluca Demartini

31

XP: opinioni personali

- ◆ E' molto difficile mettere in pratica tutte le pratiche!
- ◆ La migliore è il testing
- ◆ Esistono committenti che mettono a disposizione un loro dipendente a tempo pieno?
- ◆ L'XP developer ha molte più soddisfazioni
- ◆ Se l'XP funziona, sviluppare software è molto più piacevole



Gianluca Demartini

32

Bibliografia

- ◆ Kent Beck, "Programmazione estrema Introduzione" Addison-Wesley, 2000 ISBN 88-7192-088-0



- ◆ www.extremeprogramming.org
- ◆ www.jera.com/techinfo
- ◆ "Il framework JUNIT" di Mauro Lorenzutti

Gianluca Demartini

33