

Web Application Engineering

sviluppo agile delle applicazioni Web

cristian lucchesi
IIT-CNR

Pescara, 15-16 Maggio 2007
AleI – Ud'A



agenda

- Web Engineering, Web Application, Web development
- I modelli agili di sviluppo del software
- Analisi dei requisiti
- SCM, pianificare le attività, controllare il processo di sviluppo
- Progettazione e pattern

Agenda

cont.

- Framework e componenti ad alto livello
 - ORM - Java Persistence API
 - Java Server Faces
 - JBoss Seam
- Evolvere verso la prossima generazione:
Ajax

Web Application

- In software engineering, a Web application or webapp is an application that is accessed with a Web browser over a network such as the Internet or an intranet. Web applications are popular due to the ubiquity of the browser as a client, sometimes called a thin client. The ability to update and maintain Web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity. Web applications are used to implement Webmail, online retail sales, online auctions, wikis, discussion boards, Weblogs, MMORPGs and many other functions.

From Wikipedia, the free encyclopedia

Web Application & Web development

- in letteratura spesso ci si riferisce a Web Site, Web-based applications, Web-based systems e altri varianti come **Web Application**
- noi utilizzeremo il termine Web application per rappresentare tutte queste varianti. In ogni caso hanno qualche scopo informativo e aiutano le persone a compiere qualche attività
- il termine **Web development** è utilizzato per descrivere lo sviluppo, la messa in produzione ed il mantenimento delle Web Application

Percezione del Web development

6. Web project planning and management

5. Web-based system

4. Web site construction

3. Web site design

2. Web page design

1. Web page construction

- il primo livello è quello più visibile e si riferisce alla realizzazione della pagina HTML (e dei CSS)
- il secondo e il terzo si riferiscono all'interazione con l'utente e fanno parte dell'area della Human Computer Interaction
- i livelli 4,5,6 sono quelli che necessitano di un approccio ingegneristico
- lo sviluppo delle applicazioni web richiede un approccio integrato e sistematico di professionalità diverse

Caratteristiche Web Application

1. tempi di sviluppo richiesti molto brevi	2. specifica dei requisiti insufficiente
3. "content is king", contenuti integrati inextricabilmente con il software, aggiornati molto frequentemente	4. costante evoluzione con brevi cicli di revisione
5. piccoli team (spesso di giovani) che lavorano con scadenze molto brevi	6. utilizzate da tipologie di utenti molto diversi con differenti requisiti, percezioni e conoscenze
7. molta attenzione alla soddisfazione dell'utente	8. utilizzo di tecnologie e metodologie emergenti
9. grande varietà di device e di formati da utilizzare e supportare	10. criticità nelle prestazioni
11. standard in continua evoluzione a cui l'applicazione si deve adeguare (nuove tecnologie, nuovi linguaggi, nuovi tool,...)	12. consapevolezza della necessità di molte discipline come graphic design, information architecture, HCI, ...
13. importanza della sicurezza e dalla privacy	14. problematiche legali, sociali e etiche

Alei/Ud'A - Pescara, 15-16 maggio 2007 - cristian lucchesi, IIT-CNR

7

Semplici vs Complesse

Semplici	Complesse
contenuto delle informazioni statico	pagine web dinamiche, contenuti che cambiano spesso
poche informazioni principalmente testuali	molte informazioni (anche multimediali)
semplice navigazione	difficoltà di navigare e trovare le informazioni
accessi poco frequenti o limitata utilità limitata interattività e funzionalità	integrata con database o altri sistemi di pianificazione, schedulazione attività e tracking
le prestazioni non sono un requisito fondamentale	high performance e alta affidabilità richieste
sviluppata da un singolo individuo o un team molto piccolo (1-2) persone	necessità di un team di sviluppo con conoscenze in aree diverse
pochi cambiamenti nel tempo	pronte anche per grandi evoluzioni
facili da creare	necessitano il controllo di molti componenti e l'integrazione di tecnologie e metodologie anche molto diverse
principalmente utilizzate come presenza sul web, non come un mezzo di comunicazione interattivo	necessitano della gestione del progetto e di metodologie per la gestione del ciclo di vita

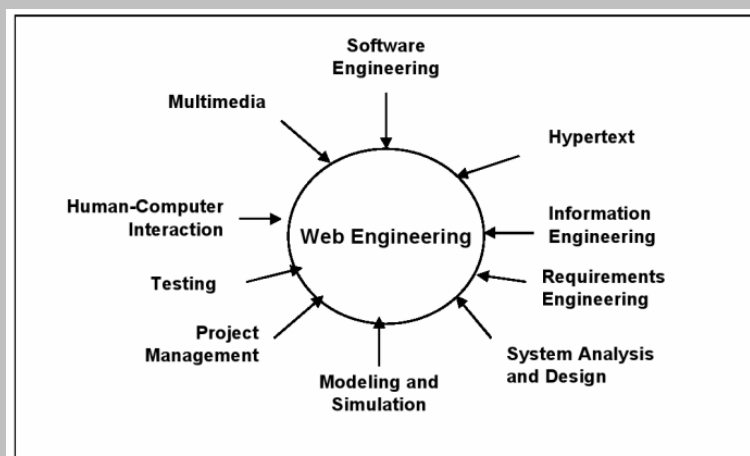
Alei/Ud'A - Pescara, 15-16 maggio 2007 - cristian lucchesi, IIT-CNR

8

Web Engineering

- Web Engineering is the application of systematic, disciplined and quantifiable approaches to development, operation, and maintenance of Web-based applications (Murugesan et al, 1999) It (Web Engineering) is a response to the early chaotic development of Web sites and applications as well as recognition of a divide between Web developers and conventional software developers (Murugesan et al, 2002)

Web Engineering - Multisciplinarietà



Web & Software development

- ci sono diversi punti in comune tra i due processi di sviluppo, che includono:
 - la necessità di metodologie
 - gestione dei requisiti
 - la programmazione
 - testing e mantenimento del software
- Web Engineering ha molto da imparare dall'Ingegneria del Software anche se alcuni metodologie devono essere adattate alle diverse caratteristiche

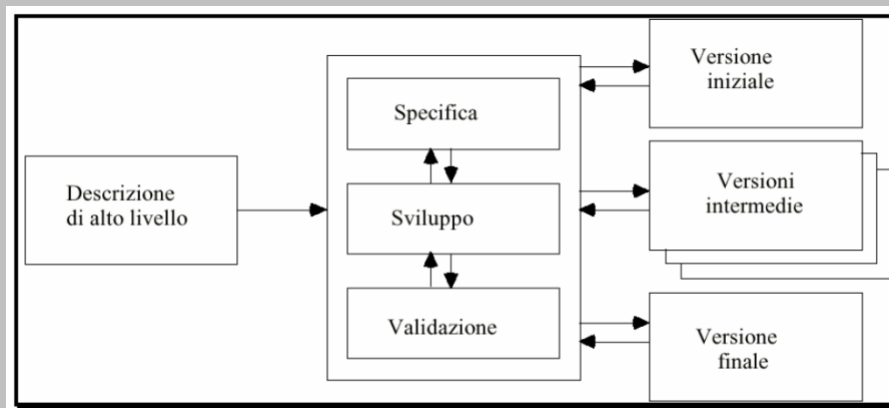
Fallimento dei progetti Web-based

- uno studio del Cutter Consortium del 2000, evidenziava seri problemi nella realizzazione di progetti largamente basati sul Web
 - i tempi di consegna non rispettavano le necessità del business nel 84% dei casi
 - i ritardi nelle scadenze contagiavano il progetto nel 79% delle volte
 - i progetti sforavano il budget previsto nel 63% dei casi
 - sistema finale che non rispettava le funzionalità richieste nel 53% dei casi
 - sistema finale di bassa qualità nel 52% dei casi
- la Web Engineering cerca di risolvere queste problematiche attraverso un approccio sistematico e disciplinato

Web development process

- il processo di Web development individua i passi e le attività dello sviluppo del sistema
- definisce una serie di passi che gli sviluppatori possono seguire, valutando i risultati
- il processo di sviluppo deve:
 - tener conto della complessità del sistema
 - tener presente la modificabilità dei requisiti nel tempo
 - permettere di acquisire i requisiti non sempre facilmente disponibili
 - favorire l'integrazione tra differenti discipline e conoscenze
 - facilitare la comunicazione tra gli appartenenti al team
 - supportare continue evoluzioni ed il mantenimento del software
 - aiutare la gestione della complessità e diversità del processo di sviluppo rispetto ai sistemi tradizionali
- un processo di tipo evolutivo aiuta a gestire i progetti Web-based


Processo evolutivo



Agile development

- Fine anni 90' diverse metodologie ottengono sempre maggiore attenzione, ognuna ha combinazioni di vecchie e nuove idee.
- Tutte le metodologie enfatizzano degli aspetti comuni:
 - stretta collaborazione tra programmatori e esperti di business
 - comunicazione face-to-face (più efficiente della documentazione scritta)
 - rilascio frequente di nuove versioni
 - gruppi di programmatori snelli e auto-organizzati
 - modalità di organizzare codice e sviluppatori tale che l'inevitabile cambio di requisiti non sia critico

Metodologie di sviluppo Agile

- queste metodologie impongono un processo che ha l'obiettivo di sviluppare software prevedibile e efficiente.
- nel febbraio 2001 a **Snowbird**  in Utah si incontrano per trovare un terreno comune 17 persone rappresentanti di diverse metodologie di sviluppo software
 - eXtreme Programming
 - DSDM
 - SCRUM
 - ...
- Il gruppo che si forma è conosciuto come Agile Alliance www.agilealliance.org

Manifesto per Agile Software - Principi

- I principi su cui si basa una metodologia leggera che segua i punti indicati dall'Agile Manifesto, sono solo quattro:
 - le persone e le interazioni sono più importanti dei processi e degli strumenti (ossia le relazioni e la comunicazione tra gli attori di un progetto software sono la miglior risorsa del progetto)
 - è più importante avere software funzionante che documentazione (bisogna rilasciare nuove versioni del software ad intervalli frequenti, e bisogna mantenere il codice semplice e avanzato tecnicamente, riducendo la documentazione al minimo indispensabile)
 - bisogna collaborare con i clienti al di là del contratto (la collaborazione diretta offre risultati migliori dei rapporti contrattuali)
 - bisogna essere pronti a rispondere ai cambiamenti più che aderire al progetto (quindi il team di sviluppo dovrebbe essere autorizzato a suggerire modifiche al progetto in ogni momento)

eXtreme Programming - XP

- eXtreme Programming incorpora le più importanti idee proposte dall'approccio Agile allo sviluppo di software
- riguarda non solo aspetti tecnici ma anche aspetti sociali alla programmazione.
- XP è stato sviluppato da Kent Beck e Ward Cunningham
Kent Beck "Extreme programming explained: Embrace change" (Beck, 1999)
- XP is a mechanism "for social change, a style of development, a path to improvement, an attempt to reconcile humanity and productivity and a software development discipline" (Beck, 2004).

Extreme Programming - valori

- **Semplicità**
 - design semplice e codice molto semplice possono essere facilmente compresi da tutto il gruppo di lavoro
- **Comunicazione**
 - costruire e disseminare conoscenze tra gli sviluppatori e i partecipanti al team (compreso i clienti), creando e condividendo una visione comune di come il sistema funziona
- **Feedback**
 - è strettamente collegato alla comunicazione e alla semplicità.
 - il Feedback è relativo a diversi elementi: feedback dal sistema, feedback dal client, feedback dal gruppo di lavoro

Extreme Programming – valori

cont.

- **Coraggio**
 - il coraggio emerge in diversi aspetti correlati alla programmazione come assicurare ai programmatori di sentirsi a loro agio nel rifattorizzare il loro codice se necessario
- **Rispetto**
 - rispetto può essere inteso in diverse accezioni: la considerazione del lavoro dei partecipanti al gruppo, la convinzione dello sviluppo di un prodotto di elevata qualità o la ricerca del miglior design per la soluzione

Extreme Programming – principi

XP principi:

- fare test prima di programmare a mantenere funzionanti tutti i test
- integrare e testare l'intero sistema molte volte, anche molte volte al giorno
- produrre tutto il software in coppia, due programmatori di fronte ad uno schermo
- iniziare progetti con un semplice design che si evolve costantemente, riduce la complessità non necessaria e aumenta la flessibilità
- produrre una parte piccola del sistema velocemente e farla crescere per provare il valore dell'applicazione

(Beck, 1999)

Alei/Ud'A - Pescara, 15-16 maggio 2007 - cristian.lucchesi@iit.cnr.it

21

XP Core Practice

- XP è definito dalle pratiche usate.
- Le pratiche variano nel tempo e a seconda del progetto in cui vengono utilizzate:
 - Planning the game
 - Simple Design
 - Pair Programming
 - Testing
 - Refactor
 - Short releases
 - Coding Standard

Alei/Ud'A - Pescara, 15-16 maggio 2007 - cristian.lucchesi@iit.cnr.it

22

XP Core Practice: planning the game

- sviluppo dell'applicazione accompagnato dalla stesura di un piano di lavoro
- piano definito e aggiornato a intervalli brevi e regolari dai responsabili del progetto, secondo le priorità aziendali e le stime dei programmatori
- i programmatori partecipano, in modo attivo, alla pianificazione
- la pianificazione coinvolge sia utenti responsabili del progetto che sviluppatori per stabilire un equilibrio dinamico fra le esigenze di tutti

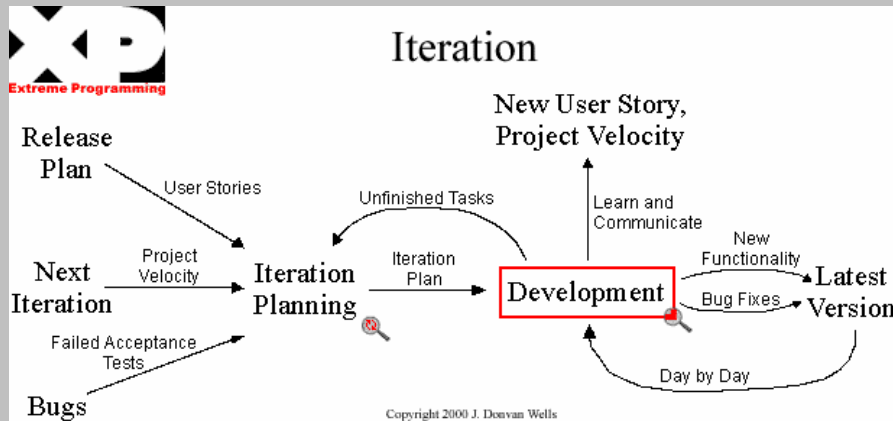
XP Core Practice: planning the game

cont.

- gli utenti finali dell'applicazione presentano gli obiettivi da raggiungere descrivendo una serie di scenari (storie)
- gli sviluppatori stimano il tempo necessario per la realizzazione di ogni storia
- le storie vengono ordinate da utenti e responsabili secondo la loro priorità di realizzazione, dopo che gli sviluppatori ne hanno stimata la rispettiva difficoltà
- dalla sintesi delle valutazioni i responsabili del progetto generano la pianificazione delle attività, intesa come l'insieme di storie che dovranno essere realizzate per il prossimo rilascio e le date previste

XP Core Practice: planning the game

cont.



Alel/Ud'A - Pescara, 15-16 maggio 2007 - cristian.lucchesi, IIT-CNR

25

XP Core Practice: simple design

- la struttura dell'applicazione deve essere la più semplice possibile
- l'architettura del sistema deve essere comprensibile da tutte le persone coinvolte nel progetto
- non devono esserci parti superflue o duplicazioni
- le parti che compongono il sistema devono essere, soltanto, quelle strettamente necessarie alle esigenze correnti
- solo quando nuove circostanze lo richiederanno, verranno progettati nuovi componenti, eventualmente riprogettando anche quelli già esistenti

Alel/Ud'A - Pescara, 15-16 maggio 2007 - cristian.lucchesi, IIT-CNR

26

XP Core Practice: pair programming

- la scrittura vera e propria del codice è fatta da coppie di programmatori che lavorano al medesimo terminale
- le coppie non sono fisse, ma si compongono associando migliori competenze per la risoluzione di uno specifico problema
- il lavoro in coppia permette, scambiandosi periodicamente i ruoli, di mantenere mediamente più alto il livello d'attenzione
- i locali dove si svolge il lavoro devono permettere senza difficoltà di lavorare a coppie

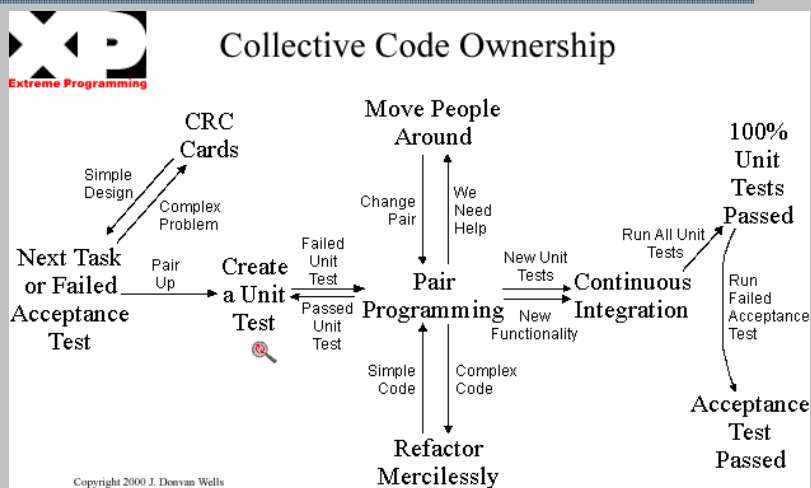
XP Core Practice: testing

- ogni funzionalità va sottoposta a verifica, in modo che si possa acquisire una ragionevole certezza sulla sua correttezza
- test di sistema costruiti sulla base delle storie concordate con il committente
- test di unità che devono poter essere rieseguiti automaticamente, con tempi dell'ordine dei minuti
- ogni ristrutturazione o modifica del codice deve mantenere inalterato il risultato dei test già considerati
- i test vengono, generalmente, scritti prima della codifica della funzionalità

XP Core Practice: refactor

- soprattutto dopo molti cambiamenti nel tempo il codice diventa poco maneggevole
- i programmatori spesso continuano a utilizzare codice non più mantenibile perché continua a funzionare
- quando stiamo rimuovendo ridondanza, eliminiamo funzionalità non utilizzate e rinnoviamo un design obsoleto stiamo rifattorizzando
- il refactoring mantiene il design semplice, evita complessità inutili, mantiene il codice pulito e conciso così che sia facilmente comprensibile, modificabile e estendibile

XP Core Practice: collective code ownership



Riferimenti

- Web Engineering: Introduction and Perspectives" by San Murugesan and Athula Ginige Chapter 1 in "Web Engineering: Principles and Techniques" (Suh, W. ed.), Idea Group Publishing, 2005
http://www.idea-group.com/downloads/excerpts/01_20Suh.pdf
- International Conference on Web Engineering 2007:
<http://icwe.com.polimi.it/>
- Extreme Programming: A gentle introduction
<http://www.extremeprogramming.org/>
- XProgramming.com - an Agile Software Development Resource
<http://www.xprogramming.com/>

