



Assignment User Stories

Processo e sviluppo del software

Mauro Manfredelli 781266

UNIVERSITÀ DEGLI STUDI DI MILANO - BICOCCA

INDICE

1	Assunzioni	2
2	Users stories.....	3
2.1	Prima storia	3
2.1.1	Caratteristiche.....	3
2.2	Seconda storia	4
2.2.1	Caratteristiche.....	4
2.3	Terza storia	5
2.3.1	Caratteristiche.....	5
2.4	Quarta storia	6
2.4.1	Caratteristiche.....	6
2.5	Quinta storia.....	7
2.5.1	Caratteristiche.....	7
2.6	Indipendenza delle storie	8
3	Analisi dei costi.....	9
3.1	Stima delle storie.....	10

1 ASSUNZIONI

Considero Facebook come un'applicazione già implementata su cui andare ad implementare nuove funzionalità (modifiche evolutive) descritte dalle *User stories*. L'analisi dei costi verrà effettuata tramite **Funcion points (FP)**, per questo ipotizzo di avere dei processi già stimati tramite FP relativi, per esempio, ai casi d'uso dell'analisi iniziale (per semplicità non ci sarà aggiornamento degli FP dei processi iniziali); all'applicazione sarà associato un certo numero di FP calcolato dai vari processi che lo compongono, da cui potrò derivare le informazioni necessarie all'analisi. Per poter stimare i tempi in personhours, a ogni membro del team sarà associato un numero di FP giornalieri e il costo totale della User story sarà dato dal rapporto tra il numero di FP della User story e il numero di FP giornalieri del singolo; considerando 8 ore lavorative al giorno derivo il costo in *personhours*.

2 USERS STORIES

Verranno presentate 5 possibili User stories relative a Facebook; ogni storia deve rispettare le seguenti caratteristiche: *independent, negotiable, valuable for users, assessable, small, and testable*.

L'indipendenza verrà discussa alla fine di questo capitolo, nella sezione 2.6.

Per l'implementazione di queste storie ipotizzo che su Facebook sia già implementato il caricamento di immagini o foto da galleria o memoria.

2.1 PRIMA STORIA

Caricare le foto scattate al momento con la fotocamera dello smartphone, tramite applicazione.

2.1.1 Caratteristiche

- **NEGOTIABLE**

Nella storia non sono specificati dettagli funzionali, i dettagli più rilevanti sono stati descritti come *NOTE*.

NOTE: La foto può essere in un formato qualsiasi senza vincoli di qualità.

- **VALUABLE FOR USER**

Non sono stati inseriti dettagli implementativi nella storia; viene descritta una funzionalità che agevolerà l'utente nell'utilizzo del sistema in situazioni reali.

- **ASSESABLE**

Il caricamento di una foto è facilmente gestibile con una interfaccia grafica e una base di dati che tenga traccia dei dati; è necessario che l'utente permetta all'applicazione di accedere alla fotocamera per lo scatto della foto.

- **SIZE**

Il tempo di implementazione per questa storia poteva essere alto nel caso in cui fossero stati incluse più possibilità di caricamento (oltre a foto scattate, anche foto provenienti da qualsiasi altra applicazione installata), ma avendo ristretto il campo, si riesce a rispettare questa caratteristica. Molte caratteristiche sono già implementate e possono essere riutilizzate nello stesso modo anche in questa storia.

- **TESTABLE**

È importante che l'interfaccia sviluppata sia usabile per l'utente e che resti valida per aggiornamenti successivi.

Acceptance test:

1. L'utente accede alla fotocamera dall'applicazione;
 2. Scatta una foto;
 3. La foto viene mostrata lato interfaccia; l'utente può aggiungere un nome e/o una descrizione;
 4. L'utente salva la foto che viene pubblicata sul profilo.
-

2.2 SECONDA STORIA

Cercare le foto presenti in un profilo tramite parola chiave o data di caricamento.

2.2.1 Caratteristiche

- **NEGOTIABLE**

Nella storia non sono specificati dettagli funzionali, i dettagli più rilevanti sono stati descritti come *NOTE*.

NOTE Può essere conveniente limitare il numero di risultati della ricerca per evitare di aumentare i tempi di attesa.

NOTE per UI: Un solo campo di ricerca che filtra su entrambi.

- **VALUABLE FOR USER**

La storia riguarda il filtraggio di dati resi pubblici da un utente sul suo profilo, che possa facilitare la ricerca di determinati contenuti.

- **ASSESSABLE**

Non c'è nessuna scrittura di dati, sono sufficienti i dati presenti; la sequenza dei passi da implementare è semplice: imputazione del campo, ricerca e filtraggio.

- **SIZE**

L'implementazione è semplice e richiede poco più di un giorno.

- **TESTABLE**

Deve essere estendibile per poter permettere la ricerca anche su più campi se necessario e deve avere buone performance e tempo di risposta in modo da non lasciare l'applicazione appesa per troppo tempo.

Acceptance test:

1. L'utente accede alla sezione delle foto caricate e imputa la parola da cercare;
 2. Il sistema cerca le foto sui campi per cui è permessa la ricerca
 3. Il sistema filtra i risultati e li mostra lato interfaccia
-

2.3 TERZA STORIA

Cambiare la visibilità delle foto senza rimuoverle dal profilo.

2.3.1 Caratteristiche

- **NEGOTIABLE**

Nella storia non sono specificati dettagli funzionali, i dettagli più rilevanti sono stati descritti come *NOTE*.

NOTE: Quattro livelli di visibilità:

- *Solo io: visibili solo all'utente che le ha caricate;*
- *Amici: visibile solo all'utente e agli amici presenti nella sua lista di amici;*
- *Amici di amici: come Amici più gli amici presenti nella lista amici degli amici;*
- *Tutti: visibili a tutti.*

- **VALUABLE FOR USER**

Viene descritta un'opzione che permette la tutela della privacy dell'utente.

- **ASSESABLE**

La modifica è ristretta alle sole immagini: è facilmente gestibile con l'aggiunta di un campo per il salvataggio e controllo all'accesso.

- **SIZE**

Anche se per questa storia sarà necessario modificare la parte dati, a livello di application logic resta semplice e sviluppabile in una giornata e mezza.

- **TESTABLE**

Deve essere modificabile in qualsiasi momento. Lo scopo principale è tutelare la Privacy di ogni utente per le immagini.

Acceptance test:

1. L'utente sceglie la foto e ne modifica la visibilità a "solo io";
2. Il sistema salva la nuova visibilità; da questo momento solo il profilo proprietario della foto può visualizzarla.

2.4 QUARTA STORIA

Aggiungere un commento a una foto.

2.4.1 Caratteristiche

- **NEGOTIABLE**

Nella storia non sono specificati dettagli funzionali, i dettagli più rilevanti sono stati descritti come *NOTE*.

NOTE: Chi può aggiungere un commento? Dipende dalle impostazioni di privacy del profilo a cui appartiene l'immagine.

- **VALUABLE FOR USER**

All'utente interessa far sapere a un suo amico cosa pensa di una sua immagine, in maniera simile all'aggiunta del "mi piace".

- **ASSESABLE**

È necessario legare alle immagini un nuovo nodo per poter salvare il commento, e il riferimento a chi l'ha scritto e a quale foto. È sufficiente la tastiera dello smartphone per l'imputazione.

- **SIZE**

È più complessa delle storie precedenti, ma comunque realizzabile in meno di una settimana. Non è prevista la risposta a un commento esistente o la gestione di più commenti alla stessa foto per migliorare quest'aspetto.

- **TESTABLE**

Per quanto detto sopra deve essere estendibile per l'aggiunta di nuove features. L'usabilità è un altro fattore importante in quanto è accessibile a tutti.

Acceptance test:

1. L'utente schiaccia sul bottone "Commenta" e si apre una casella di testo;
 2. L'utente scrive il commento e lo salva;
 3. Il sistema salva il commento e lo mostra sotto la relativa foto.
-

2.5 QUINTA STORIA

Menzionare tramite tag le altre persone presenti in una foto.

2.5.1 Caratteristiche

- **NEGOTIABLE**

Nella storia non sono specificati dettagli funzionali, i dettagli più rilevanti sono stati descritti come *NOTE*.

NOTE: Le persone menzionate devono per forza essere presenti nella lista di amici dell'utente.

NOTE per UI: Può essere utile aggiungere una funzione per il riconoscimento delle aree delle immagini, per esempio in corrispondenza di un viso, per l'aggiunta di un tag.

- **VALUABLE FOR USER**

È di valore per l'utente perché permette a diversi utenti di condividere la stessa fotografia senza doverla caricare più volte. Un utente che non è abituato a caricare le sue, avrà comunque delle foto in cui è presente un tag a lui relativo.

- **ASSESABLE**

La funzione che riconosce i visi delle persone può essere implementata seguendo alcuni di questi algoritmi: PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), metodi Kernel, metodo delle wavelet Gabor, modello di Markov nascosto.

- **SIZE**

La realizzazione di questa storia può risultare laboriosa nel caso in cui la realizzazione di un algoritmo di riconoscimento facciale non sia immediata. In questo caso potrebbe essere separata in una storia separata. A livello di analisi iniziale, però, resta implementabile in meno di una settimana.

- **TESTABLE**

Dovrebbe essere riusabile per poter permettere l'aggiunta di un tag anche in altre situazioni.

Acceptance test:

1. L'utente preme il bottone "Aggiungi Tag";
 2. Sceglie l'area dell'immagine a cui mettere il tag;
 3. Sceglie l'amico da una lista;
 4. Salva la selezione aggiungendo il tag.
-

2.6 INDIPENDENZA DELLE STORIE

Le cinque storie possono essere implementate indipendentemente: avendo assunto che sia già gestito il caricamento delle immagini o foto sull'applicazione, si possono implementare le storie in un qualsiasi ordine. "Caricare una foto scattata dallo smartphone" utilizzerà molta logica del caricamento di un'immagine qualsiasi; una volta caricata sarà gestibile esattamente come un'immagine, quindi si possono implementare prima le storie che arricchiscono la gestione dell'immagine e poi questa.

Stesso ragionamento può essere fatto per le altre storie: si può implementare l'aggiunta di un commento e poi l'aggiunta di un tag o viceversa; si può implementare prima la ricerca, e così via.

3 ANALISI DEI COSTI

Per l'analisi dei costi si considereranno tre aspetti dell'implementazione:

- **Application Data**: modifica della base di dati (DDL);
- **Application Logic**: modifica dell'interfaccia grafica, funzioni e procedure, classi e metodi;
- **Dependencies**: transazioni che coinvolgono la base di dati come, per modelli relazionali, INSERT, UPDATE, DELETE o SELECT; coinvolge sia l'interfaccia (inserimento di un dato, cancellazione, lettura, ...), sia la base di dati (istruzione DML).

Ogni storia sarà divisa in attività di questo tipo.

Per effettuare queste stime bisogna fare il **Benchmarking**, operazione che permette a un'organizzazione che raccoglie e analizza propri dati, di creare una **IT performance baseline**, utilizzata per stimare *produttività, qualità, costo e effort*.

Per fare il Benchmarking e ottenere la baseline, utilizzo i dati aggiornati al 2009 del *QSM Business, Engineering e Real Time* provenienti da oltre 13.000 progetti completati.

Size: FP	Duration (Months)	Effort (PM)	Average Staff (FTE)	FP/PM
50	5.7	11.9	2.0	7.1
100	6.4	17.9	2.7	8.6
250	7.6	31.6	4.1	10.8
500	8.5	46.4	5.6	13.1
1,000	9.0	71.0	7.4	15.5
Min: 10	1.6	1.6	0.4	1.1
Max: 5,000	42.2	1,705.0	121.4	234.0

BASELINE

Ipotizzando una size di 1000 FP, derivo 15,5 FP/PM (Function Points / Person-month). Questo vuol dire che ogni membro del mio team produrrà 15,5 FP al mese considerando una size iniziale di 1000 FP (progetto medio/grande); considerando un mese di 20 giorni, togliendo i giorni non lavorativi, derivo **0,775 FP al giorno**.

Dalla tabella emerge un altro dato utile: **Average Staff** (personale medio); il team sarà mediamente composto da 7,4: nel caso di storie grandi si può pensare di utilizzare due persone insieme.

Dividendo l'implementazione in analisi, sviluppo e test, si può ipotizzare di impiegare tre persone nell'implementazione. Non lavoreranno tutti e tre contemporaneamente alla storia (i test non possono essere fatti durante la fase di analisi, così come gli sviluppi), ma una o due persone alla volta; non viene fatta distinzione tra analisi, sviluppo e test per l'assegnazione dei FP, dunque le storie verranno implementate con una velocità di **0,775 FP/PM (Function point / person-day)**.

Questo sarà il dato che utilizzerò per stimare i *persondays* di lavoro. Considerando 8 ore di lavoro al giorno, si ottengono **0,097 FP/PH**.

3.1 STIMA DELLE STORIE

Caricare le foto scattate al momento con la fotocamera dello smartphone, tramite applicazione.	
Attività	FP
Nessuna modifica alla base di dati: le foto verranno salvate nello stesso modo già implementato per il caricamento di immagini da galleria o altro	0
Nuovo bottone che permetta l'accesso alla fotocamera e il caricamento della foto scattata.	1,5
Salvataggio della foto	0,5
Lettura della foto caricata, già presvista per le immagini	0
Totale	2

Molti aspetti sono già stati implementati per il caricamento di immagini senza l'integrazione con la fotocamera e possono essere riutilizzati per questo caso.

Circa 2,5 *person-days* o 20,6 *person-hours*.

Cercare le foto presenti in un profilo tramite parola chiave o data di caricamento	
Attività	FP
Nessuna modifica alla base di dati, la ricerca verrà fatta su campi già presenti	0
Aggiunta di una barra di ricerca nella sezione foto	0,5
Lettura dei dati in base ai campi "Descrizione" e "Data caricamento" dell'immagine	1
Totale	1,5

Anche qui nessuna modifica alla base di dati.

1,93 *person-days* o 15,46 *person-hours*.

Cambiare la visibilità delle foto senza rimuoverle dal profilo.	
Attività	FP
Aggiunta di un campo per memorizzare la visibilità, di default a "Tutti".	0,5
Aggiunta di un menù per la scelta del cambio di visibilità	0,5
Scrittura della visibilità scelta	0,5
Lettura della visibilità scelta per il controllo.	0,5
Totale	2

Sono presenti attività per tutti i tre gruppi.
2,5 *person-days* o 20,6 *person-hours*.

Aggiungere un commento a una foto.	
Attività	FP
Aggiunta di un nodo per il salvataggio di commenti.	1
Aggiunta di un bottone con casella di testo per l'imputazione, con controllo della privacy	0,5
Scrittura del commento	0,5
Lettura dello storico dei commenti associati a una foto	1
Totale	3

L'attività più complessa è a livello di *application data*: aggiunta di un nodo relazionato alle foto.
3,87 *person-days* o 30,92 *person-hours*.

Menzionare tramite tag le altre persone presenti in una foto.	
Attività	FP
Aggiunta di un nodo per i riferimenti ai profili dei tag	1
Aggiunta di un menu a tendina per selezionare il profilo da menzionare	0,5
Implementazione di una funzione per il riconoscimento delle aree a cui aggiungere il tag	1,5
Scrittura del tag	0,5
Lettura dei tag presenti su una foto	0,5
Totale	4

5,16 *person-days* o 41,23 *person-hours*.

Risulta essere un'implementazione più complicata delle altre.

Si può pensare di assegnare quest'implementazione a due persone: una persona potrebbe concentrarsi solo su "Implementazione di una funzione per il riconoscimento delle aree a cui aggiungere il tag", in modo da separare il carico in 1,5 FP per il riconoscimento facciale e 2,5 FP per il resto. Altrimenti si potrebbe dividere la storia in due storie più piccole dove una prevede solo l'aggiunta di un tag e l'altra il riconoscimento facciale.

Mantenendo la storia unita e separando l'implementazione su due persone (1,5 FP per uno e 2,5 per l'altro) il tempo si riduce a 3,22 *person-days* o 25,77 *person-hours*.