

Progettazione partecipativa e co-costruzione con gli utenti: la metodologia degli Interactive Use Case

Vincenzo D'Andrea¹, Diego Calzà², Gianni Jacucci³

Università degli Studi di Trento

[vincenzo.dandrea@unitn.it; gianni.jacucci@unitn.it; diego.calza@know-change.com]

Sommario

La *co-costruzione* di sistemi informativi con gli utenti può essere realizzata con l'ausilio degli "*Interactive Use Case*" (caso d'uso interattivo, in breve *IUC*), uno strumento pittorico ed interattivo, posizionato tra lo *Use Case* (la descrizione del "caso d'uso") ed il *Mock-Up* (il modello non funzionante con valore metaforico). Esso può fornire il contesto ed il supporto pratico per un dialogo tra sviluppatori ed utenti di un sistema informativo nelle Metodologie Partecipative. Per contestualizzare lo strumento, descriviamo anche un particolare caso di studio, riguardante una azienda che chiameremo KN, in cui lo *IUC* è stato usato con successo nel processo di analisi di un sistema informativo complesso, permettendo una comprensione condivisa della complessità del problema.

Introduzione

Introdotta in Scandinavia negli anni Ottanta, basata sulla *co-costruzione* con gli utenti e sulla loro *appropriazione* della tecnologia, la progettazione partecipativa è la metodologia ineludibile per lo sviluppo di successo di *infrastrutture informative orientate al lavoro*.

In ambiti sociali in cui gli aspetti umani di conoscenza, apprendimento ed organizzazione prevalgano sugli aspetti scientifico-tecnici, le applicazioni del computer mostrano sempre particolari esigenze e dinamiche, seguono destini peculiari non sottostanti a logiche deterministiche, e necessitano di una epistemologia diversa per essere realizzate e spiegate. Queste applicazioni richiedono in definitiva una disciplina tutta propria per essere inquadrare, comprese e gestite: *l'informatica sociale*.⁴

Nel regno dell'informatica sociale, il lavoro di realizzazione dei servizi informativi non può essere fatto solo "in astratto", nel senso cioè di limitarsi a progettare e sviluppare sistemi di computer efficienti, ma esso richiede invece anche l'uso pratico degli stessi sistemi per lo scopo desiderato; per essere efficace il lavoro deve cioè essere situato nella realtà dell'azione sociale, economica e politica.

¹ Vincenzo D'Andrea è ricercatore presso il Dipartimento di Informatica e Telecomunicazioni dell'Università degli Studi di Trento

² Diego Calzà è amministratore delegato di Know-Change s.r.l., società che progetta e sviluppa sistemi informativi

³ Gianni Jacucci è professore ordinario presso il Dipartimento di Sociologia e Ricerca Sociale dell'Università degli Studi di Trento

⁴ Si veda nel lavoro di Brian Fitzgerald (1994), la discussione nel paragrafo 3.2, in cui si sottolinea l'importanza di andare oltre il punto di vista di un processo strettamente deterministico e razionalistico, affermando: «Other researchers have also questioned the validity of viewing systems development as a rational process ... System development is not just a technical process; social needs to be considered also ...»

Nell'informatica sociale, la progettazione partecipativa rappresenta un approccio moderno alla progettazione di sistemi informativi, in cui le persone destinate ad usare il sistema giocano un ruolo cruciale nel progettarlo⁵. Questa modalità di progettazione si differenzia dalla progettazione tradizionale, in quanto vede lo scopo della computerizzazione come un tentativo di fornire ai lavoratori strumenti migliori per eseguire il loro lavoro. Essa inoltre vede le applicazioni basate sul computer non in isolamento, ma piuttosto nel contesto dei luoghi di lavoro; come processi piuttosto che prodotti (Schuler e Namioka, 1993).

La progettazione partecipativa è l'approccio appropriato per la progettazione per il cambiamento e per l'uso emergente: rendere la tecnologia un attore del cambiamento iscrivendo il possibile cambiamento futuro già al momento della progettazione del sistema e rendere gli utenti attori del cambiamento tramite la loro partecipazione attiva alla progettazione.

Il nostro Laboratorio⁶ sposa l'approccio scandinavo della progettazione partecipativa allo sviluppo dei sistemi informativi e delle infrastrutture informative orientate al lavoro. Ci concentriamo sulla progettazione per il cambiamento e per l'uso emergente, sulla progettazione per il coinvolgimento dell'utente nella progettazione in uso ed infine sulla progettazione per la responsabilità sociale, accountability in inglese.

Lungo la dimensione del progettare per il cambiamento, l'informatica ha prodotto molta innovazione, consolidando metodologie di sviluppo di tipo Object Oriented, i Componenti, i Web Service.

I concetti di responsabilità sociale della tecnologia e di coinvolgimento dell'utente nella progettazione in uso richiedono che progettisti, utenti e tecnologie impiegate si comprendano reciprocamente. La progettazione partecipativa ha quindi bisogno di linguaggi e supporti specifici di comunicazione. Ad esempio metafore di comunicazione: modelli non funzionanti degli artefatti da costruire - detti Mock-Up e gli Use Case nella analisi e progettazione della pratica UML ad oggetti - Object Oriented Analysis and Design -. Si dice che Mock-Up e Use Case vengono impiegati come oggetti di frontiera - Boundary Object in inglese - come supporto alla comunicazione tra progettisti ed utenti, per permettere il loro dialogo e facilitare la raccolta di requisiti utente corretti.

Metodologie partecipative e oggetti di frontiera

Lo scopo delle metodologie partecipative non è limitato al miglioramento della progettazione dei sistemi informativi, ma anche al prendere in considerazione gli aspetti sociali di un dato progetto, facendo attenzione specialmente ai futuri utenti del sistema.

Citando Brian Fitzgerald (1994):

Boiland (1979) argues that organisational problem situations do not exist as an independent reality but require human interpretation, a point also raised by Davies & Ledington (1991). This is dependent on the people involved and as Checkland (1984) points out, "uniformity of

⁵ A questo proposito, si vedano i lavori di Blomberg, e Kensing (1998), di Ehn e Kyng (1991) e di Ehn e Sjørgen (1991).

⁶ Si tratta del Laboratorio di Ingegneria Informatica e Analisi Organizzativa, che fa parte del Dipartimento di Sociologia e Ricerca Sociale dell'Università degli Studi di Trento. Il Laboratorio è a sua volta collegato ad alcune *spin-off* che hanno avuto origine al suo interno. Tra queste Know-Change, con cui è stata condotta la ricerca-azione qui descritta.

perception cannot be imposed on autonomous human beings". The ingenuity and ability of the developer cannot be compounded into any development methodology.

Gli utenti finali sono considerati attori di primaria importanza sia dal punto di vista sociale sia dal punto di vista del sistema. Essi vengono attivamente coinvolti in ogni attività progettuale dai progettisti del sistema – progettisti che si curano quindi sia delle pratiche di lavoro che della costruzione del prodotto.

Vi sono due concetti fondamentali che governano l'implementazione nelle metodologie partecipative. Il primo è l'apprendimento reciproco in cui utenti e sviluppatori insegnano l'un l'altro pratiche di lavoro e potenzialità tecnologiche, condividendo un'esperienza comune. Il secondo è il progettare tramite il fare.

Secondo Kensing e Munk-Madsen (1993), non c'è una tecnica infallibile per ottenere il coinvolgimento attivo degli utenti. Essi sostengono che il problema reale è un problema di comunicazione; la difficoltà sta quindi nel trovare un linguaggio comune per iniziare il dialogo e per stabilire un contesto di progettazione.

Se i sistemi debbono essere sviluppati con lo stesso linguaggio astratto usato per la loro progettazione, allora c'è il rischio di creare un gap tra il sistema implementato ed il bisogno reale. Ci sono due possibili uscite a questo dilemma: (1) trovare una via per creare specifiche migliori e più accurate e (2) intraprendere evoluzioni multiple, procedendo per tentativi ed errori (per esempio, con prototipi).

Secondo Fitzgerald (1994):

Other approaches to systems development, such as evolutionary development and prototyping, have emerged in response to some of the inadequacies of the waterfall life-cycle, particularly the suggestion that requirements can be specified in advance. These approaches are characterised by the evolutionary nature with which the system is produced in an iterative fashion, perhaps through a series of prototypes (cf. e.g. Agresti, 1986; Davis et al., 1988; Mayhew and Dearnley, 1987; Mansuy, 1989).

In un gran numero di casi, lo sviluppo del software si basa su specifiche formali o informali di requisiti. L'uso delle specifiche è legato ad un modus operandi analitico, tipico degli sviluppatori, inteso a ridurre la complessità per mezzo dell'astrazione. Gli utenti trovano spesso difficoltà nel chiudere il gap tra una descrizione astratta dell'uso e la conoscenza e le competenze professionali che caratterizzano l'uso stesso (Grønbaek, Kyng e Mogensen, 1993). Per superare la difficoltà di avere a che fare con linguaggi diversi, impieghiamo gli "oggetti di frontiera" (Boundary Objects in inglese), cioè strumenti che permettono a due realtà diverse di comunicare chiaramente ed efficacemente.

Nel nostro approccio gli Interactive Use Case assumono il ruolo di oggetti di frontiera. Secondo Leigh Star e Griesemer (1989, pag. 393). E' della massima importanza che persone appartenenti a comunità di pratica diverse siano capaci di usare un oggetto di frontiera in termini di ciò che Chrisman (1999) chiama "un punto di riferimento comune" nelle conversazioni. La principale proprietà che caratterizza un oggetto di frontiera è l'essere capito da entrambe le parti. Si usano quindi gli oggetti di frontiera come "media di coordinamento e allineamento" (Fischer e Reeves, 1995).

Metodologie partecipative nel ciclo di vita di un sistema informativo

Nelle varie fasi del ciclo di vita del sistema, procedendo verso lo sviluppo e la manutenzione del sistema, il formato ed il linguaggio di rappresentazione dello Use Case assumono caratteristiche differenti.

I piani descritti qui di seguito sono una rappresentazione metaforica delle quattro fasi della nostra metodologia partecipativa per l'analisi e lo sviluppo del sistema.

Il *piano del problema* contiene modelli riguardanti il problema affrontato dal software. Questi modelli riguardano ciò con cui il software avrà a che fare ma che non potrà modificare. In questa fase gli sviluppatori “apprendono” dagli utenti; in questa fase partecipativa usiamo descrizioni dello Use Case espresse in semplice testo. Con questo strumento semplice gli utenti possono esprimere le loro pratiche di lavoro e le parti critiche di rilievo di queste pratiche.

Il *piano dell'analisi* contiene modelli che descrivono, in termini dell'area di dominio, in che modo ci proponiamo di risolvere i problemi identificati nel piano del problema. Qui gli utenti imparano dagli sviluppatori. In questa fase usiamo gli Interactive Use Case; ciò ci permette di identificare la discrepanza tra ciò che il sistema dovrebbe fare e ciò che la soluzione corrente ci propone – lo IUC può essere qui descritto come uno strumento per identificare correttamente i bisogni dell'utente.

Il *piano del progetto* contiene modelli che descrivono come la soluzione proposta sarà realizzata. In questa fase usiamo i Mock-Up. Essi permettono anche agli utenti di indicare eventuali discrepanze tra ciò che il sistema promette e ciò che esso in realtà realizza.

In fine, il *piano della implementazione* è il codice vero e proprio – in che modo il progetto è reso concreto in un particolare ambiente di programmazione.

Interactive Use Case

Tra gli strumenti della progettazione partecipativa adottati dal nostro Laboratorio, mettiamo in risalto gli Interactive Use Case, dandone una definizione ed una collocazione anche in termini temporali d'impiego rispetto agli altri che li precedono o li succedono.

Use Case

Il ruolo tradizionale degli Use Case è quello di descrivere il comportamento di un sistema in risposta agli input degli utenti (Jacobson et al., 1992). Considerando il ciclo di vita di un sistema come descritto dalle metodologie orientate agli oggetti, gli Use Case sono utilizzati nella fase di raccolta dei requisiti. Nel nostro approccio partecipatorio, proponiamo sia per Use Case sia per Interactive Use Case un ruolo come oggetti di confine (Boundary Objects) tra sviluppatori ed utenti di sistemi informativi.

Il significato letterale di Use Case è “caso d'uso”, ovvero una modalità di utilizzo del sistema. Per creare uno Use Case la prima cosa da fare è identificare gli attori, umani e non-umani, che utilizzeranno il sistema. Vanno capiti e definiti quali siano i loro obiettivi nell'utilizzo del sistema, e quali siano gli scenari in cui tali obiettivi si possono realizzare. Uno scenario è una sequenza di passi, simile ad una sceneggiatura, in cui si identificano gli input degli attori e le corrispondenti azione e risposte del sistema. Uno Use Case è un particolare di uno scenario, un particolare del disegno complessivo. Gli Use Case vengono

generati a partire da uno o più requisiti utente, ovvero cosa gli utenti richiedono al sistema in termini di funzioni e comportamenti.

Con gli Use Case si assume una prospettiva di tipo “black box”, la scatola nera che non può essere aperta, con cui viene data importanza al solo comportamento del sistema.

Gli strumenti generalmente utilizzati per rappresentare gli Use Case sono sia testuali che in forma di diagrammi, che mostrano una rappresentazione visuale dell'utilizzo previsto per il sistema. Ad esempio, la metodologia UML, ovvero Unified Modeling Language (Rumbaugh, Jacobson, e Booch, 1998), assegna un ruolo importante e ben definito agli Use Case, prescrivendo strumenti e metodi per crearli.

Mock-Up

Un Mock-Up è un artefatto che rappresenta un sottoinsieme delle funzionalità di un sistema informativo.

Le motivazioni per la costruzione di un Mock-Up possono essere di varia natura, in questo ambito ci interessa sottolinearne l'uso come veicolo per il coinvolgimento degli utenti. Rivisitare insieme agli utenti, con l'aiuto di Mock-Up, le funzionalità di un sistema in via di costruzione è un momento importante della nostra metodologia partecipatoria.

Nella visione di Bödker and Grönbaek (1991), il Mock-Up è considerato uno strumento per il cosiddetto approccio esplorativo, e viene utilizzato sia per simulare un sistema sia come prototipo da gettare dopo l'uso. Un approccio esplorativo proposto da questi autori è il Cooperative Prototyping (ovvero Prototipazione Cooperativa) in cui utenti e progettisti di un sistema cooperano attivamente nella progettazione, interagendo mediante prototipi. Dal punto di vista del Participatory Design si tratta di una situazione di reciproco apprendimento, in cui utenti e progettisti possono apprendere gli uni dagli altri (Rumbaugh, Jacobson e Booch, 1998).

Interactive Use Case

Gli Interactive Use Case (in breve IUC) sono derivati dagli Use Case in forma descrittiva e diagrammatica, come strumenti per l'analisi dei sistemi informativi. L'obiettivo è fornire all'utente una possibilità di esprimere il proprio modo di lavorare con il sistema futuro. Questi strumenti vengono usati durante le sessioni di analisi, che coinvolgono progettisti ed utenti, per creare una conoscenza comune e condivisa.

Uno IUC incorpora la logica alla base del sistema ed una descrizione delle parti che lo comporranno. Esso viene costruito con uno strumento che produca un risultato graficamente simile alle schermate del sistema futuro, tuttavia lo scopo principale degli IUC non è dare una anteprima della grafica del sistema, ma mostrarne la logica. Sono quindi importanti i raggruppamenti di categorie di dati (che devono risultare congrui con l'uso previsto) e i pulsanti che attiveranno azioni del sistema. Dati e azioni devono appartenere ad uno stesso Use Case o anche ad una sua parte. Alle schermate dello IUC possono essere sovrapposte note e commenti. Nello IUC non sono previste funzionalità, l'unica azione automatizzata dovrebbe essere il passaggio da uno IUC ad un altro (oppure tra parti diverse dello stesso IUC) mediante pulsanti presenti nelle schermate dello IUC. Un aspetto importante dell'uso degli IUC è il modificarli, o anche produrli, durante le sessioni di lavoro con gli utenti del futuro sistema.

Gli utenti vengono coinvolti anche nella scelta tra diverse opzioni di sviluppo del sistema, vedendole “in azione” durante le sessioni di lavoro con gli sviluppatori. Anche i requisiti

non funzionali che emergono in queste sessioni (come per esempio, regole di accesso o relative alla sicurezza del sistema) possono essere facilmente integrate come annotazioni nella rappresentazione grafica degli IUC.

Gli IUC possono essere realizzati con una molteplicità di strumenti, da programmi per realizzare presentazioni a pagine HTML, a veri e propri ambienti di programmazione.

Il caso di studio

L'azienda KN (il nome non è quello reale) è una azienda di consulenza che opera nel settore della gestione e riduzione dei costi per l'acquisto di beni, ad esclusione dei materiali per la produzione. I consulenti KN definiscono per il cliente gli scenari di acquisto più convenienti, in accordo con le necessità del cliente stesso. KN lega quindi il proprio successo ai risultati ottenuti in termini di risparmio per i clienti.

I consulenti KN oggi sono i depositari di tutta la conoscenza necessaria per definire i migliori scenari di acquisto. Gli scenari dipendono da un gran numero di variabili, sia di natura misurabile o razionale (come per esempio: categorie di prodotti, politiche di acquisto definite, tendenze di mercato, promozioni temporanee, ...) che non (variazioni nel management, ingresso di nuovi attori nel mercato, ...).

In questo momento, KN cerca di consolidare le proprie procedure operative ed il modello su cui si fonda, sfruttando nel contempo le potenzialità offerte da Internet verso nuovi modelli operativi. L'azienda cerca di fare un passo oltre la semplice idea di utilizzare la rete per distribuire il know-how tra i consulenti. L'obiettivo è stabilire relazioni di lungo periodo con i clienti, fornendo sia i modelli di acquisto ottimali, sia i sistemi per mantenere questi modelli costantemente aggiornati e ottimizzati. In questo modo, i clienti potrebbero utilizzare i modelli di KN all'interno delle proprie procedure, spostando il ruolo dei consulenti KN verso quello di certificatori di prodotti e processi.

KN ha tentato più volte la strada di un pacchetto applicativo standardizzato. Nel caso più recente, l'idea è stata concentrarsi sul cuore del business di KN: incaricare un consulente in grado di certificare la validità delle decisioni di acquisto di una azienda, mantenendo relazioni stabili nella catena del valore. Il sistema era centrato sui prodotti acquistati, organizzati ad albero, separando i beni in base al livello di costo. L'obiettivo di questa strutturazione era evidenziare il risparmio in funzione delle differenze di costo. Questa idea apparentemente valida è stata tuttavia la causa dell'insuccesso di questo prodotto – i prezzi variano troppo rapidamente e focalizzare tutto su questo aspetto ha prodotto un sistema troppo instabile per gli scopi prefissati.

Obiettivo di KN con l'avvio di un nuovo progetto di sistema informativo è stato il mettere a punto una procedura di business innovativa, per mezzo di un sistema di gestione della conoscenza e di supporto alle decisioni. Nel sistema da realizzare, KN chiede la possibilità:

- di rappresentare il prodotto e tutte le variabili che producono le migliori condizioni di acquisto (in termini di attributi del prodotto stesso o in relazione ad altri prodotti);
- definire e la rappresentare regole operative (business rules) che stabiliscano i criteri per identificare le migliori condizioni di acquisto;
- di consentire l'accesso e la gestione via rete della knowledge base dei prodotti e delle regole per confrontarli in benchmark – in questa modalità di lavoro sono necessarie adeguate misure di sicurezza e controllo degli accessi;
- di generare scenari di miglior acquisto, individuati caso per caso accedendo alla knowledge base ed alle regole operative;

- di archiviare gli scenari, per costruire una storia dei casi trattati da KN.

Definiti gli obiettivi di massima dell'azienda, prima di intraprendere lo sviluppo di un nuovo sistema è utile chiedersi quali siano state le ragioni dell'abbandono dei precedenti approcci. Tra le ipotesi, le seguenti sono particolarmente rilevanti:

- non erano (e non sono) disponibili dei sistemi informativi standardizzati capaci di rispondere alle necessità dell'azienda;

- le richieste di KN sono troppo elevate, non è realistico pensare di ottenere tali risultati con un sistema informativo, la complessità del sistema non è gestibile, i costi sono troppo elevati;

- gli sviluppatori delle precedenti soluzioni non sono stati in grado di identificare correttamente i requisiti, funzionali e non, di KN;

- KN ha necessità di essere aiutata a descrivere e a rappresentare i propri bisogni:

KN continua ad evolvere, operando in un mercato dinamico, ed aggiorna quindi di continuo i propri modelli operativi.

Per rispondere a queste problematiche, il nostro approccio è stato l'adozione di metodologie partecipative. Si è formato un gruppo di lavoro, scegliendo con attenzione i partecipanti. Hanno partecipato al gruppo di lavoro analisti senior di KN, manager contabili, manager di vendite, ciascuno con le proprie attese e dubbi rispetto al nuovo tentativo di costruzione di un sistema informativo.

Nei primi incontri, il gruppo di lavoro ha ricostruito in termini di Use Case le principali procedure operative di KN. Lo staff del nostro laboratorio, dopo aver presentato la metodologia, ha principalmente operato con un ruolo di mentore. Nel seguito delle attività, si è passati da Use Case ad Interactive Use Case. Il fuoco dell'attività si è progressivamente spostato sui processi gestiti dal sistema informativo.

Conclusioni

Nel caso KN, lo strumento Interactive Use Case è stato introdotto dopo aver raggiunto uno stato stabile degli Use Case.

Raggruppando le sessioni di lavoro in classi ognuna con particolari finalità, si possono distinguere:

- *il cosa*: disegnare a video l'interazione possibile tra persone e macchina, nel momento dell'informatizzazione dei processi di lavoro. Il focus è stato posto sugli input che l'utente deve fornire al sistema e sugli output che questo deve restituire in seguito all'elaborazione. Mai si è affrontato in questa fase il come ciò possa essere poi implementato in forma tecnologica;

- *il come*: raffinare il progetto delle interfacce, curando le sequenze logiche dei vari passaggi d'interazione persona-macchina (seguendo in prima battuta quanto definito nel flusso degli eventi degli Use Case). Il focus è stato posto sulla scelta accurata degli elementi d'interfaccia più idonei alla realizzazione di una interazione efficace. (menu, finestre accostate, cartelle, moduli e sottomoduli, etc.)

- *la verifica*: svolta utilizzando la tecnica del "walking on the IUC", ovvero percorrendo lo IUC insieme agli utenti, per simulare assieme l'interazione con l'applicazione, finalizzando l'attività verso l'individuazione dei punti di miglioramento.

- *la documentazione*: rimuovere le ambiguità tra gli elementi d'interfaccia attraverso la documentazione visiva di ogni oggetto grafico riportato.

Lo IUC fissa i bisogni di KN ed evidenzia anche la zona di requisiti in continua fluttuazione; è la prima volta che KN dispone della “fotografia” del proprio sistema informativo. Inoltre KN diviene consapevole delle parti che ormai si sono definite in dettaglio e di quelle ancora in continua evoluzione. Questo ha portato alla decisione, condivisa con KN, di dividere la progettazione futura in due diverse fasi: la fase relativa alla progettazione delle componenti ritenute maggiormente stabili (quelle che interessano la porzione consolidata dello IUC) con attività di progettazione ed implementazione attivabili da subito senza rischio di dispersione di risorse economiche; e la fase d’esplorazione approfondita delle componenti instabili (quelle che interessano la porzione dello IUC ancora “fluttuante”). Va evidenziato che la fluttuazione di una parte dello IUC non è dovuta all’insufficiente raccolta di requisiti, o ad un’analisi incompleta: KN ha compreso che questa parte di sistema informativo deve essere progettata e realizzata per l’evoluzione e non per la stabilità. Questo perché i modelli di rappresentazione della conoscenza basati sui benchmark di acquisto possano essere adattati dall’utente (o meglio dal consulente KN) durante l’uso, a seconda del caso specifico. Si tratta di una operazione di adattamento su misura quasi da sartoria, detta infatti tailoring. Quindi le componenti di sistema relative alla parte fluttuante, da adattare, dovranno essere progettate per il cambiamento e non per la stabilità (design for change). E’ questa probabilmente la reale criticità che in precedenza ha reso difficile l’individuazione di soluzioni accettabili per KN.

In conclusione, si può affermare che gli obiettivi stabiliti per la fase d’analisi del sistema informativo di KN sono stati raggiunti. I risultati positivi di questo lavoro consistono:

- nell’aver creato una visione, condivisa da sviluppatori ed utenti, del problema da affrontare; entrambe le parti si sono rese pienamente conto di dove sono gli aspetti più problematici e di dove bisogna investire ulteriormente per realizzare un sistema che si adatti alle esigenze di KN;

- nell’aver condotto la fase di mutuo apprendimento del design partecipativo, utilizzando gli Use Case come “cavallo di battaglia”; KN ha inoltre manifestato interesse ad utilizzare gli UC come linee guida per il Sistema Qualità. Durante questa fase di analisi partecipata, si sono osservati dei continui scostamenti rispetto alle idee iniziali, il drift (Ciborra, 1996), questi scostamenti sono stati tradotti all’interno degli IUC per poi poter ricostruire il percorso che ha portato alla struttura finale del sistema informatico.

- nell’aver fornito a KN una prima soluzione “visibile” ed “interattiva” del sistema informativo meta dei loro bisogni;

KN oggi riconosce in forma chiara i propri bisogni in termini di sistema informativo aziendale, ed è consapevole di cosa va fatto per affrontarli e risolverli.

Bibliografia

- Agresti W. (1986), *New Paradigms for Software Development*, IEEE Computer Society Press, Washington DC.
- Blomberg J. e Kensing F. (1998), prefazione al numero speciale su Participatory Design, *Computer Supported Cooperative Work*, 7: 163-165.
- Bødker S. e Grønbaek K. (1991), “Design in Action: From Prototyping by Demonstration to Cooperative Prototyping”, in *Design at work: cooperative design of computer systems*, edito da Greenbaum J. e Kyng M., Lawrence Erlbaum Associates, Hillsdale, pp. 197-218.
- Boland R. (1979), “Control, causality and information systems requirements”, *Accounting, Organizations and Society*, 4: 259-275.

- Cantoni F., Negri D. e Perondi E. (2004), "Processo di sviluppo del Sistema Informativo Lavoro (SIL) della Regione Emilia Romagna", in
- Checkland P. (1984), "Systems theory and information systems", in *Beyond Productivity: Information Systems Development for Organisational Effectiveness*, edito da Bemelmans T., Elsevier Science Publishers B.V., North Holland Press, pp. 9-21
- Chrisman N. (1999), *Trading Zones or Boundary Objects: Understanding Incomplete Translations of Technical Expertise*, 4S, San Diego.
- Ciborra, C. (1996), *Groupware and Teamwork*, Wiley, Chichester.
- Davies L. e Ledington P. (1991), *Information in Action: Soft Systems Methodology*, Macmillan Press, London.
- Davis A., Bersoff E. e Comer E. (1988), "A strategy for comparing alternative software development life cycle models", *IEEE Transactions on Software Engineering*, 14 (10): 1453-1460.
- Du  R.T. (2002), *Mentoring Object Technology Projects*, Yourdon Press.
- Ehn P. e Kyng M. (1991), "Computer: Mocking-it-up or Hands-on the Future" in *Design at work: cooperative design of computer systems*, edito da Greenbaum J. e Kyng M., Lawrence Erlbaum Associates, Hillsdale, pp. 169-196.
- Ehn P. e Sj rger D. (1991), "From System Description to Scripts for Action", in *Design at work: cooperative design of computer systems*, edito da Greenbaum J. e Kyng M., Lawrence Erlbaum Associates, Hillsdale, pp. 241-268.
- Fischer G. e Reeves B.N. (1995), "Creating Success Models of Cooperative Problem Solving", in *Readings in Human-Computer Interaction: Toward the Year 2000*, 2nd edition, edito da Baecker R.M. et. al., Morgan Kaufmann Publishers, San Francisco, pp. 822-831.
- Fitzgerald, B. (1994) "The Systems Development Dilemma: Whether to Adopt Formalised Systems Development Methodologies or Not?", in *Proceedings of Second European Conference on Information Systems*, edito da Baets W., Nijenrode University Press, Holland, pp. 691-706.
- Gr nbaek K., Kyng M. e Mogensen P. (1993), "CSCW Challenges: Cooperative Design in Engineering Projects", *Communications of the ACM*, 36 (4): 67-77.
- Jacobson I., Ericsson M., Jacobson A. e Overgaard G. (1992), *Object-Oriented Software Engineering: A Use case Driven approach*, ACM Press, New York,
- Kensing F. e Munk-Madsen A. (1993), "Participatory Design: Structure in the Toolbox", *Communications of the ACM*, 36 (4): : 78-85.
- Leigh Star S. e Griesemer J.R. (1989), "Institutional Ecology, 'Translations', and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology 1907-39", *Social Studies of Science*, 19: 387-420.
- Mansuy J. (1989), "Evolutionary development strategy for MIS", *Journal of Systems Management*, July, pp. 7-13.
- Mayhew P. e Dearnley P. (1987), "An alternative prototyping classification", *Computer Journal*, 30 (6): 481-484.
- Rumbaugh J., Jacobson I. e Booch G. (1998), *The Unified Modeling Language Reference Manual*, Addison-Wesley.
- Schuler D. e Namioka, A. (1993), *Participatory Design: Principles and Practices*, Lawrence Erlbaum Associates.

Rumbaugh J., Jacobson I. e Booch G. (1998), "Open-Ended Interaction in Cooperative Prototyping: a video-based analysis", *Scandinavian Journal on Information Systems*, 3: 63-86.