

Unità 8

Promuovere la qualità

Livello 3 – approfondimento

Temi trattati all'interno dell'Unità

- Requisiti funzionali e non funzionali.
- Gli stakeholders.
- Strumenti di modellazione
- L'approccio di qualità "good enough".

Sommario

CAPIRE I FABBISOGNI: COMUNICARE, MODELLARE, CONDIVIDERE	1
LA GESTIONE DEL PROGETTO SI FA AGILE: LA QUALITÀ GOOD ENOUGH	5
BIBLIOGRAFIA.....	6

CAPIRE I FABBISOGNI: COMUNICARE, MODELLARE, CONDIVIDERE

Come si sta lavorando? Come si può lavorare? Cosa serve? Condividete la soluzione? Sono le domande con cui ci si confronta sempre quando si **analizza** una possibile soluzione, soprattutto nella realizzazione di un sistema informatico o nell'uso di tecnologia digitale.

Identificare fabbisogni e definire correttamente i requisiti della soluzione sono attività determinanti per il successo di un progetto.

Per raggiungere questo obiettivo bisogna:

- **Comunicare** per colmare il divario (**gap**) tra cultura digitale e dominio in cui opera la soluzione
- **Modellare** in modo semplice e intuitivo il contesto in cui si opera
- **Condividere** con tutti gli **Stakeholder** il disegno di una soluzione

Il **Requisito** è una condizione o una caratteristica alla quale la soluzione deve essere conforme: è osservabile dall'esterno, soddisfa un bisogno, uno standard o altri vincoli che vengono imposti.

La raccolta, la definizione e l'analisi dei requisiti forniscono una descrizione della soluzione da realizzare.

L'analisi definisce i modelli di lavoro funzionali e i modelli dei dati trattati.

Raccogliere i Requisiti significa:

- ✓ Definire il problema
- ✓ Comprendere i bisogni
- ✓ Identificare le caratteristiche del sistema
- ✓ Redigere una documentazione di "Definizione dei Requisiti"

Precisando meglio, un requisito è l'aspetto di un certo contenuto o comportamento che risulta essere necessario o desiderato dal cliente.

I requisiti poi sono caratterizzati dal tipo di bisogno identificato, bisogno definito da due assiomi di base:

- Cosa la soluzione **deve fare**.

- Cosa la soluzione **deve avere**.

Il *fare* è correlato direttamente alle funzioni che ci si aspetta dalla soluzione pensata, *l'aver* individua caratteristiche e proprietà della soluzione.

Di conseguenza:

- I **requisiti funzionali** sono le cose che la soluzione **deve fare**:
 - sono raccolti dall'apprendimento del lavoro,
 - individuano la parte del lavoro che sarà svolta dalla soluzione.
- I **requisiti non-funzionali** sono le proprietà che la soluzione **deve avere**:
 - Descrivono le proprietà o caratteristiche come:
 - Aspetto e Sensazione (*Look and Feel*).
 - Usabilità.
 - Prestazione.
 - Operabilità.
 - Portabilità e manutenibilità.
 - Sicurezza.
 - Aspetti culturali e sociali.
 - Aspetti legali.

Un vincolo invece è un requisito che influenza tutto il prodotto e spesso viene determinato da norme, leggi e quant'altro.

I requisiti, durante la loro raccolta, devono essere documentati e condivisi. Documentare i requisiti significa:

- ✓ Creare una base di comunicazione tra le parti coinvolte.
- ✓ Definire gli input necessari al team di progetto.
- ✓ Definire gli input necessari al test e collaudo.
- ✓ Fornire i contenuti nell'accordo contrattuale fra le parti.
- ✓ Sapere cosa fa, e quindi permettere il controllo dell'evoluzione del sistema prodotto.

Quest'ultimo punto significa definire lo scopo del progetto, ovvero definire l'obiettivo, scrivendo una chiara e concisa frase che descriva cosa si vuole che il prodotto faccia e quali vantaggi dovrà portare, ad esempio ridurre i costi, semplificare le operazioni, incrementare le entrate, fornire un servizio migliore, agevolare la compatibilità con la legge.

Come si misurano i vantaggi?

I requisiti devono essere misurabili e verificabili. La misura deve essere correlata al bisogno e al miglioramento che ci aspettiamo di ottenere nell'applicazione del requisito. Nella stessa definizione, quindi, deve essere indicata al meglio la misura di riscontro e verifica dell'applicazione del requisito, descrivendo il vantaggio atteso sia in fase di collaudo sia in fase di reale applicazione nell'ambiente in cui viene integrata la soluzione.

In questa fase di definizione bisogna anche capire se l'obiettivo è ragionevole, ovvero:

- ✓ È avviabile?
- ✓ Tutti gli *stakeholders* lo vogliono? È condiviso?
- ✓ È fattibile?
- ✓ È raggiungibile?

Questo aspetto implica una comunicazione e condivisione della raccolta dei requisiti con tutti i soggetti coinvolti: gli *stakeholder*. Chi sono?

Come abbiamo visto nelle unità precedenti uno *stakeholder* è chiunque abbia interesse in una determinata soluzione. Nel lavoro di raccolta dei requisiti è sempre meglio effettuare un'analisi per produrre la lista di tutti gli *stakeholder* coinvolti.

Non identificare tutti gli *stakeholder* significa infatti sia tralasciare alcuni requisiti, sia ritrovarsi in situazioni potenzialmente conflittuali, soprattutto se alcuni requisiti impattano negativamente su soggetti che non sono stati coinvolti.

Tra gli *stakeholder* troviamo sicuramente il **cliente**

Il cliente è chi paga per lo sviluppo della soluzione; diventa quindi determinante la domanda: cosa produrre per soddisfare il proprio cliente?

Il cliente può essere l'ufficio commerciale, l'utente della direzione o un rappresentante di un potenziale mercato.

Uno stakeholder importante è "lo sponsor".

Lo sponsor è chi spinge per richiedere la soluzione e la sponsorizza, ovvero se ne fa promotore. È in qualche modo un garante, ed è auspicabile che abbia l'autorità di coordinare tutti coloro che lavorano sul progetto. Lo sponsor è responsabile della fornitura delle risorse necessarie: risorse umane, spazio negli uffici, accesso alle informazioni, e strumenti. Lo sponsor deve avere, ed è importante che abbia, l'autorità di decidere in caso di controversie fra requisiti di *stakeholders* diversi; spesso lo sponsor può essere lo stesso cliente.

Ma, soprattutto tra gli *stakeholder*, troviamo **gli utenti**.

Gli utenti sono le persone che operano sul prodotto. Identificando gli utenti si capisce:

- ✓ il lavoro che fanno,
- ✓ la soluzione che bisogna costruire per loro,
- ✓ chi tra loro fornisce al meglio le informazioni riguardanti la funzionalità della soluzione.

Di conseguenza, ognuno degli *stakeholder* contribuisce ai requisiti secondo il proprio ruolo proprio perché uno *stakeholder* ha una sua responsabilità, la conoscenza di alcune parti del lavoro o anche un interesse in parti del lavoro futuro.

Oltre all'identificazione e al coinvolgimento degli *stakeholders*, determinante nella raccolta dei requisiti è anche la definizione del contesto del lavoro: una vera e propria delimitazione dei confini in cui si deve analizzare la soluzione.

Il **contesto** definisce l'estensione del lavoro da analizzare, ed è a sua volta definito proprio dai servizi forniti dalla soluzione analizzata e dalle necessarie interazioni con il mondo esterno.

Nel modellare il contesto iniziamo a raccogliere informazioni utili a determinare ciò che viene richiesto dalla soluzione.

Il diagramma di contesto:

- Include ciò di cui si ha bisogno per conoscere e capire il sistema da progettare.
- Indica come il sistema interviene nei cambiamenti delle attività del committente.
- Mostra i sistemi adiacenti; sistemi che non possono essere cambiati senza un accordo. I sistemi adiacenti sono anche fonti di dati per il sistema analizzato o possono a loro volta diventare clienti per i prodotti forniti dal sistema.
- Definisce le connessioni con i sistemi adiacenti e in questo modo crea i confini del sistema in progettazione.
- Definisce le funzionalità e i dati memorizzati che è necessario studiare e analizzare per il sistema in analisi.

Altro elemento importante se non essenziale, oltre alla definizione del contesto, è dato dalle parole ovvero dal **glossario**.

Costruendo il Glossario e condividendolo sia con il gruppo che lavorerà al progetto sia con gli *stakeholder* si crea il linguaggio comune che verrà usato lungo tutto il percorso della soluzione. Creare un glossario significa identificare, chiarire e usare le parole corrette e coerenti sia nell'ambito in cui viene applicata la soluzione analizzata, e quindi capire i concetti e i termini del committente, sia far comprendere al committente i termini più strettamente legati alla soluzione tecnologica e digitale in fase di implementazione.

Il **glossario** quindi raccoglie le parole (termini e definizioni) che sono utilizzate nel contesto in cui la soluzione opera.

L'utilizzo di un glossario permette:

- ✓ di non fraintendere il significato delle parole evitando i problemi che questo può creare,

- ✓ di creare una lista dei termini più importanti usati dagli stakeholder,
- ✓ di risparmiare molte ore di spiegazione,
- ✓ di raccogliere i termini del dominio applicativo analizzato.

*l'inizio della saggezza
è chiamare le cose con il loro nome
(proverbio cinese)*

Il contesto viene modellato e la modellazione aiuta a comprendere il quadro analizzato.

Nella raccolta dei requisiti e nella definizione e comprensione dei bisogni, l'uso di strumenti di modellazione favorisce la comunicazione e soprattutto la schematizzazione e semplificazione.

Perché creiamo modelli?

La creazione di un modello aiuta a capire e raccogliere i requisiti della soluzione da individuare.

Un modello semplifica la realtà, e i modelli:

- ✓ ci aiutano a visualizzare un sistema come è o come vorremmo che fosse,
- ✓ ci permettono di specificare la struttura o il comportamento di un sistema,
- ✓ ci forniscono uno schema (*template*) che ci guida nella costruzione di un sistema,
- ✓ documentano le decisioni che abbiamo preso.

Con i modelli ci focalizziamo su un solo aspetto alla volta e ogni modello può essere espresso a differenti livelli di precisione ed è utile, soprattutto in soluzioni non banali, avere un insieme di modelli, costruiti e studiati separatamente, ma strettamente interrelati.

È importante comunque avere il giusto livello di approccio nel modellare, a volte ci si può perdere esagerando nella rappresentazione.

Sono stati definiti dei linguaggi di modellazione che permettono di descrivere e utilizzare al meglio dei costrutti grafici.

I più conosciuti e praticabili, anche senza competenze tecnologiche specifiche, sono:

- l'UML (Unified Modeling Language).
- il BPMN (Business Process Modeling Notation).

L'UML è un linguaggio universale adatto a rappresentare qualunque tipo di sistema (software, hardware, organizzativo, ecc.). È uno standard dell'OMG (*Object Management Group*) dal novembre 1997 e serve a specificare le caratteristiche di un nuovo sistema oppure a documentarne uno già esistente. L'UML è al tempo stesso versatile e rigoroso, ed è quindi utile per la comunicazione tra il mondo tecnologico e il mondo più prettamente organizzativo e di business.

Il modello in linguaggio UML permette di comunicare al meglio, chiarisce cosa comunicare con il modello e a chi. Proprio per comunicare è utile costruire diagrammi semplici e accompagnarli con spiegazioni testuali. Chiunque, anche senza una grande esperienza di UML, può capire in questo modo una modellazione.

L'UML fornisce diversi diagrammi utilizzabili secondo i bisogni: diagrammi **strutturali** che permettono di definire strutture di dati (**Diagrammi delle classi e degli oggetti**), diagrammi **comportamentali** che permettono di descrivere utilizzi e flussi (Casi d'Uso, Diagrammi di Attività, Diagrammi di Sequenza).

Il *Business Process Modeling Notation* (BPMN) invece è diventato uno standard approvato nel 2006.

L'obiettivo del BPMN è di fornire una notazione comprensibile da:

- analisti che definiscono i processi,
- sviluppatori responsabili dell'implementazione tecnologica dei processi,
- persone del business che gestiranno e terranno sotto controllo i processi.

I vantaggi del BPMN sono:

- permettere alle organizzazioni di rappresentare i propri processi con una notazione intuitiva (diagramma di flusso - flow chart),
- essere uno standard che agevola la comunicazione (anche nei confronti di organizzazioni esterne).

Il limite del BPMN è che permette di modellare solo processi e flussi, e quindi non è utile per rappresentare altri aspetti importanti come:

- strutture organizzative, ruoli, responsabilità,
- scomposizioni funzionali,
- modelli di dati e informazioni,
- regole di business.

LA GESTIONE DEL PROGETTO SI FA AGILE: LA QUALITÀ GOOD ENOUGH

L'agilità nel digitale misura l'avanzamento della realizzazione di una soluzione basandosi su funzionalità importanti operative e operanti fin dall'inizio, di volta in volta perfezionate e arricchite con nuove funzioni.

Questo permette di gestire progetti con il vantaggio di potersi fermare prima del completamento totale, quando almeno l'80% del valore è stato creato, e massimizzando in questo modo il lavoro non fatto (viene detto sviluppo incrementale *Feature Driven* – guidato dalle funzionalità realizzate).

Gli sviluppi tradizionali (*blackbox*) generavano valore e soluzione solo al completamento finale del progetto.

L'approccio **Agile** permette invece di trovare soluzioni in tempi ravvicinati e risparmiare costi.

I cambiamenti sono tollerati fin quando è possibile: il valore per il cliente è centrale. Ed elemento essenziale, anche nella gestione agile del progetto, è la qualità. Risparmiare sui costi va bene, ma bisogna anche riuscire in ogni caso a soddisfare il cliente e la sua aspettativa nella soluzione.

Il *good enough* (**abbastanza buono**) si fonda su una visione utilitaristica della qualità. Ovvero: **le possibili conseguenze positive superano le potenziali conseguenze negative**. Il *good enough* è una alternativa alle metodologie che miravano alla migliore qualità possibile, spesso definita *zero defect* (zero difetti).

Il *good enough* è diventato una pratica normale nel nostro quotidiano anche se forse non ce ne rendiamo conto o non sappiamo che facciamo scelte direttamente discendenti da questo approccio.

La frase "*non è perfetto ma può bastare alle mie necessità*" è collegata anche ai due assunti "*non è perfetto ma ho quello che mi serve*" e "*costa poco e non fa perdere tempo*". Dal mondo tecnologico il *good enough* ha rapidamente, come il digitale, pervaso le nostre abitudini.

Il principio del *good enough* ha plasmato la società con profonde trasformazioni. Proprio nelle tecnologie, oggi, non ha senso ricercare la perfezione: se l'obiettivo fosse sempre e solo la perfezione non sarebbe nato il mondo delle *start-up*.

Tutta l'innovazione tecnologica si poggia sul *good enough*. I tempi sono sempre così ristretti che non si immettono sul mercato prodotti definitivi ma spesso versioni beta, che non vuol dire scadenti, ma solo migliorabili.

Il miglioramento avviene per approssimazioni, soprattutto coinvolgendo e interagendo con i consumatori del prodotto. Il modello di business associato al principio del *good enough* si fonda sulla collaborazione e interagire con i consumatori permette di capire e soddisfare meglio le esigenze.

Il principio del *good enough* crea anche dinamiche controverse, ma anche in questo caso si tratta di averne la consapevolezza e trovare la giusta modalità d'uso.

La vera essenza del *Good Enough* non è quella di eliminare tutti i problemi indiscriminatamente, ma capire abbastanza bene problemi e benefici in modo tale da eliminare i **veri problemi** per ottenere i **veri benefici**.

Infatti oggi, nei progetti di innovazione digitale:

- Si tende a concentrarsi sulle funzionalità fornite (*features*) e sui tempi. I Project Manager vengono premiati per le funzionalità non per la qualità (*features not quality*).
- Si gestisce l'ingegneria sostenibile (*sustained engineering*)
Non c'è mai abbastanza tempo per fare bene subito, ma c'è sempre abbastanza tempo per risolvere il problema (*fix the bugs*).

Anche nel *good enough* si applica il principio 80:20; conosciuto come Principio di Pareto in cui viene detto che il 20% delle cause, produce l'80% dei risultati. In questo caso l'80 per cento di quello che vogliamo realizzare viene raggiunto nel 20 per cento del tempo totale impiegato.

In termini pratici, se abbiamo un'ora a disposizione per fare un lavoro, in 12 minuti ne completeremo l'80 per cento e il resto del tempo lo dedicheremo al 20 per cento rimanente.

Di conseguenza dovremmo sempre porci la domanda: le funzioni (*features*) fornite superano i difetti (**defect**)? E le funzioni più importanti e attese sono nella soluzione progettata?

BIBLIOGRAFIA

TITOLO	AUTORE	EDIZIONI	ANNO
La good-enough society. Sopravvivere in un mondo quasi ottimo	Paolo Magrassi	Franco Angeli Editore	2010
UML Distilled. Guida rapida al linguaggio di modellazione standard	Martin Fowler	Pearson – Addison Wesley	2010
Mastering the Requirements Process: Getting Requirements Right (Third Edition)	Suzanne Robertson / James Robertson	Suzanne Robertson / James Robertson	2012

U4
JOB