



Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA

Concorrenza vs distribuzione nella digital industry.

Il caso Sanmarco Informatica

RELATORE

PROF. ARMIR BUJARI
UNIVERSITÀ DI PADOVA

TUTOR ESTERNO

ALEX BEGGIATO
SANMARCO INFORMATICA

CANDIDATO

DAVIDE BARASTI

DEDICA..

Abstract

...

Indice

ABSTRACT	v
ELENCO DELLE FIGURE	viii
ELENCO DELLE TABELLE	xi
1 IL CONTESTO	I
1.1 Sanmarco Informatica SpA	I
1.2 Manufacturing execution system	3
1.3 JMES	5
1.4 Le squadre JMES	6
1.5 L'interazione macchina-MES: JDI	7
1.6 Way of working	8
1.6.1 Sviluppo agile e framework Scrum	8
1.6.2 Tecnologie di sviluppo del team	8
1.6.3 Strumenti di lavoro del team	9
2 LE PROBLEMATICHE	II
2.1 Il primo sviluppo di JDI	II
2.2 I problemi con JDI	II
2.3 Il nuovo sviluppo	12
3 ANALISI, PROGETTAZIONE E SVILUPPO	13
3.1 Studio della versione precedente	13
3.2 I requisiti del progetto	13
3.3 Tecnologie e strumenti impiegati	14
3.4 Metodologia di lavoro	15
3.5 Tracer bullet development	15
4 RETROSPETTIVA	17
RIFERIMENTI	18
GLOSSARIO	19
ACRONIMI	21

Elenco delle figure

1.1	Logo BUJmes	2
-----	-----------------------	---

Elenco delle tabelle

L'avvento e la crescita di internet hanno rivoluzionato il modo di fare impresa in particolare quella industriale. Ciò che prima era un impianto isolato, con operatori e responsabili, ora è parte di una catena produttrice interconnessa che genera non solo beni tangibili ma anche dati che, se ben gestiti, possono diventare informazione utilizzata per aumentare l'efficienza dell'intera impresa in un delicato ciclo di feedback.

1

Il contesto

Prima di affrontare il problema oggetto di questa tesi è doveroso fornire una panoramica del contesto aziendale che ha caratterizzato lo stage svolto. In questo modo i capitoli successivi risulteranno più chiari e inseriti in una propria logica.

1.1 SANMARCO INFORMATICA SPA

L'azienda Sanmarco Informatica (SMI) dal 1984 si occupa di consulenza e sviluppo *software*. Si è specializzata nella progettazione, realizzazione e installazione di soluzioni a supporto dei processi aziendali di duemila aziende con numerose installazioni oltre confine.

L'azienda ha tra i suoi punti di forza quello della vicinanza ai clienti, che si traduce in diverse sedi di proprietà in Veneto, Lombardia, Emilia-Romagna e Friuli-Venezia Giulia con 450 dipendenti totali.

Il Centro Ricerca e Sviluppo (CRS) è situato a Grisignano di Zocco (VI), sede di lavoro per oltre 150 dipendenti. Questo è il fulcro dello sviluppo e della manutenzione dei prodotti *software*. Team di sviluppatori e sistemisti sono coordinati per garantire stabilità di servizio per i clienti, e un'alta personalizzazione dei prodotti offerti.

La ricerca e l'innovazione sono di grande valore per SMI, con una media del 20% di fatturato investito annualmente in ricerca e sviluppo. Sono anche numerose le risorse impiegate nella formazione e ricerca di talenti provenienti dall'Università di Padova. Un esempio di questo

è il progetto *Academy* che viene descritto nel seguente modo: "Lo scopo è quello di educare giovani allievi da inserire direttamente nel mondo lavorativo, attraverso un percorso di eccellenza che forma figure professionali altamente qualificate."*

Il CRS è anche responsabile per l'erogazione dei servizi *cloud* forniti da SMI. È infatti presente una sala server amministrata da tecnici sistemisti.

Tutta la proprietà operativa di SMI rimane in azienda, senza il bisogno di *outsourcing* ad aziende esterne.

L'azienda è organizzata in *business unit* (BU), un sottoinsieme dell'impresa che rappresenta un *business* specifico concentrato su una particolare linea di prodotti. La BU interessata dallo stage è JMES, composta da 20 persone tra sistemisti e sviluppatori, che si occupa dell'omonimo applicativo utilizzato nell'ambito della *digital industry*. Questo software serve a gestire e supervisionare l'avanzamento della produzione all'interno della fabbrica.

Figura 1.1: Logo BU Jmes



Altre BU si occupano invece di *Business Process Management* (JPA), gestione contenuti aziendali come manuali e documenti della qualità (Discovery ECM), *marketing* (nextBI), sviluppo di applicazioni e siti per aziende (4words) e molto altro.



Ulteriori informazioni riguardanti le BU o l'azienda in generale possono essere trovate sul sito ufficiale di **Sanmarco Informatica**.

*www.sanmarcoacademy.com

1.2 MANUFACTURING EXECUTION SYSTEM

L'avvento e la crescita di internet hanno rivoluzionato il modo di fare impresa in particolare quella industriale.

Ciò che prima era un impianto isolato, con operatori e responsabili, ora è parte di una catena produttrice interconnessa che genera non solo beni tangibili ma anche dati che, se ben gestiti, possono diventare informazione utilizzata per aumentare l'efficienza dell'intera impresa in un delicato ciclo di *feedback*.

La corretta gestione di questa informazione diventa fondamentale per il successo dell'impresa mantenendo alta la competitività.

Nasce quindi la necessità di un software che trasformi il dato in informazione e che faccia uso di questa per migliorare le *performance* supportando e gestendo a sua volta tutta la linea di produzione con indicazioni precise e in tempo reale.

Questo strumento prende il nome di *manufacturing execution system (MES)*.

Un software MES in pratica si occupa di raccogliere, elaborare e condividere informazioni provenienti direttamente dall'industria che normalmente restano disaggregate per diversi motivi: si trovano in formato non elettronico, sono poco precise o si trovano in ambienti isolati tra loro.

Il controllo e la gestione dell'informazione consentono non solo di mantenere un piano generale del processo produttivo aziendale, ma anche di evidenziare possibili problemi in anticipo e ottenere quindi una migliore resa di produzione.

Vediamo quali sono i principali benefici di un sistema MES[†]:

1. Riduzione di scarti e sprechi. Grazie ad una visione in tempo reale della produzione, è possibile individuare con piccoli margini situazioni in cui le unità prodotte non sono conformi, fermando la produzione e limitando gli sprechi;
2. Precisione nella definizione dei costi. Con un sistema MES, i tempi di lavoro, gli sprechi, i tempi di inattività, la manutenzione sono monitorati e registrati in tempo reale direttamente dalla fabbrica. Questo rende innanzitutto l'informazione più affidabile, e utilizzabile per riclassificare i costi e valutare criticamente situazioni con alto tasso di sprechi;
3. Ridurre i tempi di inattività. Siccome una produzione ferma difficilmente porta a un guadagno, un MES deve essere in grado di pianificare la produzione in modo che le risorse necessarie siano quelle disponibili, integrando anche i piani dei turni dei dipendenti, in modo da aumentare ulteriormente l'efficienza, riducendo quindi i costi;

[†][*Five benefits of a MES*], www.industryweek.com/companies-amp-executives/five-benefits-mes

4. Riduzione del magazzino. Le giacenze in magazzino costano, perché aumentano i costi logistici di gestione, oltre ai costi per produrre tali giacenze. Con un MES si ha uno stato aggiornato della nuova produzione, degli scarti e dei prodotti non conformi. In questo modo chi si occupa di acquisto, spedizione e pianificazione sa esattamente cosa è disponibile e cosa bisogna ordinare;
5. Riduzione di costi. Con un controllo più stretto sui tempi e i costi necessari per la produzione, è possibile snellire ulteriormente processi a supporto, come la gestione logistica del magazzino, quindi ottenendo un disimpegno di persone e attrezzature.

1.3 JMES

Nella sezione 1.2 sono stati individuati i principali benefici che l'utilizzo di un *software* MES porta.

Vediamo quali sono i benefici che JMES intende portare ai clienti con il suo prodotto:

1. Avanzamento processi produttivi. L'informazione sull'avanzamento della produzione è disponibile in tempo reale, permettendo una gestione più flessibile del lavoro. Ad esempio gli uffici commerciali che si trovano in sede distaccata dal centro produttivo possono avere informazioni dettagliate sullo stato di avanzamento di un ordine in breve tempo, senza nemmeno interpellare gli operatori o i responsabili di produzione;
2. Gestione materiali. La disponibilità in magazzino dell'ordine completato è immediatamente rilevata, permettendo un proseguimento di processo più reattivo. Si pensi ad esempio alla possibilità di richiedere una spedizione appena l'ordine risulta saldato;
3. *Monitoring* dei processi. Oltre all'avanzamento del singolo ordine, di interesse specialmente per operatori e impiegati, è possibile effettuare il monitoraggio al fine di supportare la business *intelligence* aziendale, cioè l'insieme delle strategie usate dall'impresa per analizzare i dati di produzione (stato dell'impianto, individuazione dei problemi, macchinari che portano spesso a rallentamenti);
4. Consuntivazione. Il beneficio per cui i sistemi MES nascono: rilevare i tempi di processo per valutare il discostamento da quanto preventivato. Impiegare più tempo del previsto significa ridurre il margine di guadagno che all'estremo può portare ad una perdita. Rilevare gli scostamenti permette di arrivare principalmente a due conclusioni:
 - Lo standard di valutazione è errato, ottenendo dalla serie storica dei rilevamenti una conferma che il processo produttivo non è in grado di rispettare i tempi preventivati;
 - Il processo è migliorabile. Ci sono delle casistiche che portano ad una degradazione occasionale dei tempi di produzione. Ad esempio fermi macchina ricorrenti o operatori non abbastanza efficienti.

La possibilità di rilevare queste informazioni permette di accorgersi in tempo di situazioni critiche in cui ad esempio i rilevamenti effettuati si allontanano dal piano di budget;

5. Progetto carta zero. Grazie alla gestione software della produzione, si riduce la carta circolante che molto spesso veniva utilizzata per tracciare le fasi degli impianti, per registrare le interazioni operatore-macchina e per le stampe di documenti tecnici di assemblaggio.

1.4 LE SQUADRE JMES

Come accennato nel paragrafo 1.1 esistono due squadre che lavorano in maniera sinergica per lo sviluppo e l'installazione del prodotto JMES: sviluppatori e sistemisti.

- **Sviluppatori.** Sono raggruppati all'interno del team di JMES e si occupano di implementare le funzionalità aggiuntive richieste dai clienti. Le richieste possono provenire da un singolo cliente o essere inserite in distribuzioni di aggiornamento per tutte le installazioni. La differenza sta nel valore monetario dell' *upgrade*. La figura a cui fanno riferimento è lo *Scrum Master*. All'attivo il team è composto da otto persone;
- **Sistemisti.** Effettuano un'analisi tecnica presso le sedi dei clienti evidenziando possibili problemi di compatibilità con le attrezzature presenti, propongono le possibili configurazioni del software JMES in base ai requisiti e ai vincoli posti dal cliente, organizzano l'installazione e la configurazione del prodotto. La figura a cui fanno riferimento è il capo progetto. All'attivo il gruppo è composto da dieci persone.

Per il periodo dello stage sono stato inserito nel team di sviluppo JMES.

La prima installazione di JMES risale a maggio 2018. È quindi un prodotto giovane che però è stato già apprezzato da 19 clienti attivi, ci sono inoltre 27 analisi per l'installazione in corso e altre 19 da pianificare.

1.5 L'INTERAZIONE MACCHINA-MES: JDI

Con un sistema MES tradizionale esistono due tipi di interazione: uomo-macchina e uomo-MES. La prima consiste nell'insieme di azioni che l'operatore svolge sulla macchina per avanzare nel processo produttivo, la seconda consiste nella dichiarazione delle azioni svolte dall'operatore al sistema MES.

Questo approccio può generare un bias che consiste nella differenza tra quanto effettivamente svolto durante la prima interazione e quanto dichiarato nella seconda in quanto l'operatore può accidentalmente compiere degli errori durante le dichiarazioni manuali.

Per questo il mercato ha richiesto una soluzione in grado di permettere una maggiore precisione delle rilevazioni di produzione.

L'elaborazione di questo bisogno ha portato a considerare l'introduzione di una terza interazione ai fini di ridurre quando possibile l'interazione operatore-MES.

Questa interazione è un prodotto a se stante, indipendente da JMES, chiamato JDI.

1.6 WAY OF WORKING

1.6.1 SVILUPPO AGILE E FRAMEWORK SCRUM

Indipendentemente dalle BU aziendali, la metodologia di lavoro per la gestione del ciclo di sviluppo del software è Agile, implementata con il *framework* Scrum.

Agile è una disciplina per lo sviluppo di software che pone al primo posto tra i suoi principi la soddisfazione e il coinvolgimento del cliente e la distribuzione di software funzionante in maniera regolare e a distanza di brevi periodi, dalle due settimane ai due mesi.

Il *framework* Scrum prevede di suddividere un periodo di lavoro, chiamato *sprint cycle* in tre fasi principali:

- *Planning*: il team comunica con gli *stakeholder* (rappresentati dal *product owner*), analizza e comprende i requisiti creando lo *sprint backlog*, composto di requisiti che il prodotto deve soddisfare entro la fine dello *sprint*. Questi prendono il nome di *story* se sono completabili entro uno sprint. Un insieme correlato di *Story* si chiamano *Epic*;
- *Implementation*: durante questa fase dello *sprint* il team crea delle porzioni complete di prodotto. Le funzionalità implementate in uno *sprint* provengono dallo *sprint backlog*. La durata di questa fase è, nel caso del team JMES, quattro settimane;
- *Review*: ci si riunisce per revisionare il lavoro svolto e pianificare ciò che non è stato possibile portare a termine nella fase di *implementation*. Ogni membro del team mostra una *demo* delle funzionalità sviluppate nella fase precedente;
- *Retrospective*: vengono analizzate in modo retrospettivo le fasi precedenti in un'ottica di miglioramento continuo dei processi al fine di renderli più efficienti negli sprint successivi. Si discute di strumenti e metodologie utilizzate e di come queste abbiano influito nello sviluppo. Se vengono ritenute poco utili, la loro pratica viene dismessa.

1.6.2 TECNOLOGIE DI SVILUPPO DEL TEAM

Descritte le tecnologie di sviluppo del team JMES.

Principalmente si parlerà del linguaggio Java e del framework sviluppato da sanmarco informatica per lo sviluppo: *Synergy*.

1.6.3 STRUMENTI DI LAVORO DEL TEAM

Al centro degli strumenti per la gestione del flusso di lavoro del team JMES c'è uno strumento sviluppato da IBM, Rational Team Concert (RTC). Questo offre le seguenti funzionalità:

- versionamento del codice;
- gestione dei *work item* provenienti dallo *sprint backlog* (*issue tracking system*);
- *build tool*.

RTC si integra all'interno dell'ambiente di sviluppo del team grazie a un *plug-in* completo per l'IDE Eclipse;

2

Le problematiche

2.1 IL PRIMO SVILUPPO DI JDI

L'idea di ridurre gli errori commessi durante l'interazione operatore-JMES tramite l'utilizzo di un middleware in grado di comunicare con i PLC e che effettuasse il recuper di dati che l'operatore inseriva solitamente manualmente (e.g. la conta dei pezzi prodotti da una macchina, la segnalazione di un guasto o di un fermo, ecc).

Si parla poi dello sviluppo della prima versione: poche indicazioni sulle tecnologie, nessuna restrizione sui requisiti non funzionali ecc.

Inoltre si discute della scelta architetturale che avrebbe permesso di distribuire i nodi di JDI ovunque nel mondo.

2.2 I PROBLEMI CON JDI

Si discute della mancanza di test di affidabilità e di scalabilità del prodotto sviluppato, dell'inutilità della possibilità di distribuire fisicamente JDI e dei drawback a cui l'architettura scelta portava e della scarsa manutenibilità del codice.

Di questi problemi ci si è accorti solo arrivati in produzione, cioè quando JDI è stato installato nelle fabbriche di alcuni clienti.

2.3 IL NUOVO SVILUPPO

Gli obiettivi che il nuovo JDI deve raggiungere, il cambiamento di architettura che puntava a eliminare la distribuzione del codice e che prediligeva la concorrenza.

Le garanzie di affidabilità e scalabilità ecc.

3

Analisi, progettazione e sviluppo

3.1 STUDIO DELLA VERSIONE PRECEDENTE

Nonostante il primo sviluppo di JDI non avesse avuto un risultato positivo una volta messo in produzione, le funzionalità di base di questa dovevano essere presenti anche nella nuova versione.

Come visto nel capitolo precedente, i problemi stavano nella scarsa soddisfazione di requisiti non funzionali, non nel soddisfacimento di quelli funzionali.

Parlerò poi dell'uso di librerie esterne fatto in questa versione (e.g. Google Guice, ReactiveX...)

In questa sezione si sottolineerà l'approccio di distribuzione della vecchia versione. Nelle seguenti sezioni invece apparirà chiaro come la distribuzione sia stata messa da parte sostituendola con un puro approccio concorrente.

Sarà quindi descritta la fase di analisi durata circa due settimane.

3.2 I REQUISITI DEL PROGETTO

In questa sezione vengono raccolti i requisiti estrapolati dall'analisi della vecchia versione.

Particolare attenzione sarà posta sui requisiti non funzionali. Come detto sono stati il punto debole della versione precedente.

3.3 TECNOLOGIE E STRUMENTI IMPIEGATI

Saranno qui elencate e descritte le tecnologie e gli strumenti scelti per lo sviluppo del progetto.

Saranno giustificate alcune scelte, come la riduzione quasi a zero di librerie esterne, e l'approccio "chiuso" di Sanmarco Informatica verso il fare affidamento a prodotti esterni. Ho notato infatti che un ragionamento spesso fatto dai responsabili tecnici è "preferiamo reinventare la ruota sviluppando internamente librerie anche già esistenti, per avere pieno controllo sullo sviluppo e sulla manutenzione". Cercherò di analizzare criticamente questo ragionamento.

DI SEGUITO UNA PRIMA STESURA DELLE TECNOLOGIE E DEGLI STRUMENTI IMPIEGATI.

Come vedremo nei capitoli successivi, il lavoro da me svolto non è integrato in JMES ma è un modulo esterno che amplia le sue funzionalità. Questo mi ha permesso di avere libertà nella scelta degli strumenti di lavoro, prediligendo tecnologie che ero curioso di conoscere più nel dettaglio.

I principali strumenti per lo sviluppo da me utilizzati sono stati i seguenti:

- **IDE:** IntelliJ IDEA. Un ambiente di sviluppo per Java prodotto da *JetBrains* che mi ha permesso, grazie anche a numerosi *plug-in*, di integrare in un solo ambiente lo sviluppo *software*, i test automatici, la *build* e il versionamento del codice;
- **Build tool:** Gradle. Un sistema per l'automazione di *build* che prende spunto da i classici Apache Maven e Apache Ant che permette di specificare la configurazione del progetto con un linguaggio specifico basato su Groovy.
- **VCS:** git. Sistema di versionamento distribuito utilizzato in questo progetto assieme a GitHub che ha fornito il servizio di *repository* remoto. Il suo uso è stato fondamentale in quanto sono state create, soprattutto nel primo periodo di sviluppo, diverse possibili soluzioni al problema. Git ha permesso di mantenere in parallelo ogni soluzione sviluppata, grazie all'uso di *branch*.
- **ITS:** GitHub ITS. Integrato nelle *repository* di GitHub, ha permesso di tenere traccia dei cambiamenti durante il progetto. Le *issue* sono state inserite nel *kanban board* integrata in GitHub per avere ben visibile la situazione del progetto in ogni momento della progettazione e dello sviluppo.

Per quanto riguarda invece gli strumenti che sono tornati utili durante lo sviluppo abbiamo i seguenti:

- Wireshark: strumento per la cattura e l'analisi di pacchetti utilizzato per la comprensione della comunicazione con protocollo TCP/IP tra PLC e computer.

Lo strumento per la comunicazione interna utilizzato dal team JMES è SLACK, mentre per la comunicazione più formale interna ed esterna all'azienda mi è stato fornito un account di posta elettronica.

3.4 METODOLOGIA DI LAVORO

Way of working utilizzato durante il progetto, interazione con il team JMES di cui ero parte ecc..

3.5 TRACER BULLET DEVELOPMENT

Definire questo progetto un prototipo sarebbe errato, infatti spesso questi sviluppi servono solo a dimostrare la fattibilità di un progetto e il codice utilizzato viene quasi sempre scartato. Il lavoro svolto invece è stato mirato alla creazione e all'implementazione di un'architettura solida, scalabile e affidabile della nuova versione di JDI, l'opposto quindi di un prototipo! Questo tipo di sviluppo mira a utilizzare la maggior percentuale possibile di tecnologie per stendere una prima strada verso lo sviluppo completo. Questo metodo prende il nome di *tracer bullet development*.

Si parlerà principalmente di:

- lavoro di integrazione iniziale delle tecnologie e degli strumenti utilizzati durante lo sviluppo.
- la progettazione di prototipi e l'analisi di prestazione con benchmark per testare le prestazioni prima ancora di iniziare lo sviluppo intensivo;
- la scelta del prototipo
- lo sviluppo

4

Retrospettiva

Glossario

business unit Un sottoinsieme dell'impresa che rappresenta un business specifico concentrato su una particolare linea di prodotti.. 2

digital industry Relativamente all'industria 4.0, ci si riferisce al concetto di fabbriche dove le macchine sono potenziate con connettività *wireless* e sensori, connesse a un sistema che può tenere sotto controllo l'intera filiera produttiva e compiere delle decisioni sulla base dei dati raccolti autonomamente.. 2

outsourcing L'appalto a una società esterna di determinate funzioni o servizi, o anche di interi processi produttivi.. 2

Acronimi

CRS Centro Ricerca e Sviluppo. 1, 2

SMI Sanmarco Informatica. 1, 2