Sistemi Informativi:

Il processo software

Presentazione

Mio background

Vostri obiettivi

Qualcuno ama programmare?

Domande

Argomenti principali

La natura dello sviluppo software

Il contesto e le capacita' di una azienda

Pianificazione del sistema software

Fasi del ciclo di vita del software

Approcci allo sviluppo del software

Paralleli con un SI

Come parallelo immaginiamo di progettare e costruire un singolo edificio:

Attori: cliente, impresa, abitante

Requisiti: dimensione, vani, look & feel , costi, tempi, robustezza...

Casi d'uso (funzionalita') vado dalla cucina al bagno etc.

L' impresa sceglie componenti: alcuni prefabbricati (porte..) , altri fatti ad hoc(colonne, muri) , realizza impianti che trasmettono luce acqua elettricita' etc (i dati) collega I componenti

Errori in progettazione, ritardi in realizzazione, edificio complesso, cambio requisiti da parte del cliente

Componenti fatti da persone diverse non combaciano,(= interfacce mal definite) , costose modifiche alla fine se progetto mal fatto

Paralleli con un SI

SI piu' complesso, piu' modificabile:

trasportabile/smontabile/riorganizzabile (pareti mobili, tramezzi)

complessita' invisibile dall'esterno (tanti tubi)

fragilita' (una casa non si rompe se si rompe un mattone)

Impianti ogni volta diversi e complessi: lo stato dei componenti non e' solo acceso/spento, pieno/vuoto

La natura del software (Brooks)

L’essenza del software

- Complessita’
- Conformita’
- Modificabilita’
- Invisibilita’

Creazione anziche' produzione in serie

Le variabili del software

- Partecipanti
- Processo
- Linguaggi e strumenti di definizione

Complessita’ del software (Brooks)

Da SW per matematica a SW per mondo

Visto il successo, oggi modello del mondo

Modellazione del comportamento umano, dove formalita' e matematica non aiutano

Corrispondenza esatta tra mondo reale e programma

Modelli semplici e complessi: scacchi e sistemi esperti

Complessita’ del software

nessuna coppia di componenti e’ identica

enorme numero di stati

le interazioni tra componenti crescono in complessita’ piu’ che linearmente (n**2) al crescere delle dimensioni

Il comportamento umano se scomposto in azioni elementari e' molto complesso

Conseguenze della complessita’

overhead di comunicazione, ritardi rispetto scheduling

molti stati: inaffidabilita’

funzioni complesse: bassa usabilita’

complessita’ della struttura: poca manutenibilita’

Conformita'

Il SW spesso deve modellare strutture di comportamento arbitrario create dall'uomo che non hanno nessun modello semplice

Gli scienziati sperano sempre che quel che modellano sia descrivibile tramite leggi semplici

Il computer permette di costruire modelli formali (SW) computabili molto piu' complessi di quelli scientifici pre-1950

mutevolezza del software (Brooks)

Il software evolve costantemente

- Muta la realta' attorno
- Leggi vs matematica

Sembra facile modificarlo, costo nascosto

Modifiche dopo la realizzazione, anziche' sostituzione dei manufatti

Piu' e' utile, piu' ha utenti

- si richiedono modifiche fuori dall'ambito iniziale

Invisibilita’ del software (Brooks)

Grafici Macchine a stati, Disegni 3d sono il piu' potente mezzo per modellare fenomeni ed oggetti fisici

- Ma non riescono a catturare la complessita’ dell’ esecuzione

Manca la rappresentazione temporale

- Complessita’ intuibile attraverso esecuzione
- Pensate al parallelo tra foto e video

Accidentalita' rimosse (Brooks)

Linguaggi ad alto livello eliminano dettagli inutili, con guadagni decrescenti.

Time-Sharing vs. Batch, feedback aiuta la memoria

Ambienti integrati evitano invocazione comandi separati, etc

La complessita' intrinseca e' invariata?

Fallimenti/successi (Brooks)

Fallimenti:

- Programmazione automatica
- Verifica automatica
- ~ Sistemi esperti

Successi:

- Standardizzazione /potenziamento tools(*)
- Prototipazione
- Scelta progettisti

Affrontano la coerenza requisiti col modello

(*) causata anche da declino costo HW

software e prodotti materiali

essendo immateriale, il software potrebbe evolvere a velocita' molto maggiori di un sistema fisico

- gli esperimenti e prototipi costano poco e danno subito risultati

ma:

- essendo a volte cosi' complesso ed i suoi componenti "nascosti", la sua sostituzione/modifica puo' avere impatti imprevedibili, fragilita'

Progresso dello sviluppo software

Il software e’ sviluppato tramite crescente uso di:

- Prodotti Off The Shelf (OTS) e personalizzabili
- Riutilizzo
- Prodotti di Enterprise Resource Planning
- Tecnologie a componenti: CORBA, DCOM, EJB .NET

I nuovi sviluppi sono sempre meno rispetto a quanto e' gia' stato sviluppato

Il core business rimane sviluppato ad hoc

Risorse umane nel processo

Software come sistema sociale: insufficiente l'eccellenza tecnica

2 gruppi:

- Clienti:
  - Utenti
  - Committenti
- Sviluppatori
  - Analisti
  - Programmatori
  - QA
  - Venditori
  - ....

Risorse umane nel processo

“Grandi progetti da grandi progettisti”

ambiente stimolante / motivazione

La crescente complessita’ dei SI porta a:

- Parcellizzazione / semplificazione del lavoro
- Diminuzione della creativita’ richiesta
- Obsolescenza delle competenze

I progettisti

Assumere i migliori

Formazione continua

Scambio di conoscenze, lavoro di gruppo

Rimozione degli ostacoli

Ambiente stimolante

Allineare obiettivi di tutti

Cause di fallimento

Incomprensione dei bisogni dell'utente

I requisiti cambiano troppo frequentemente

Il cliente investe poche risorse

Poca cooperazione cogli sviluppatori

Aspettative irrealistiche

Il sistema e' obsoleto gia' prima dell'uso (o troppo avanzato)

Poco coinvolgimento/ostilita' dell'utenza

Metodo di lavoro / Pocesso

Utile per:

- Ordinamento delle attivita’
- Consegna del prodotto (che cosa, quando)
- Divisione del lavoro
- Controllo, misurazione, pianificazione

Non puo’ essere codificato o standardizzato troppo rigidamente

Nella produzione di beni fisici, e' piu' facile ridurre la variabilita' delle operazioni

Dimensione del processo

il processo oggi (1)

Iterativo perche’ produce una serie di release eseguibili

Incrementale perche’ ogni modulo viene costantemente arricchito di funzionalita’

Importante individuare presto i componenti e l’ordine di sviluppo

- Moduli Coesi ed Indipendenti

Pericolo di burocrazia

il processo (2)

-ordine di esecuzione delle attivita’

-quali elaborati sono prodotti

-assegnazione delle attivita’ al team

-criteri di monitoring e misurazione

-Unified Process

strumenti CASE

Computer Aided Software Engineering

sinergico all'uso di un processo di sviluppo

- Automazione/ripetitibilita'
- Miglioramento del processo

- Puo' rendere "pigri" gli sviluppatori

Un passo indietro: la strategia

L'organizzazione di cui ci occupiamo aumenterebbe la sua efficienza con un SI?

Valutazione di una organizzazione

Capability Maturity Model (CMM)

Libro da Carnegie Mellon nel 1995

finanziato da US DoD per valutare i fornitori SW

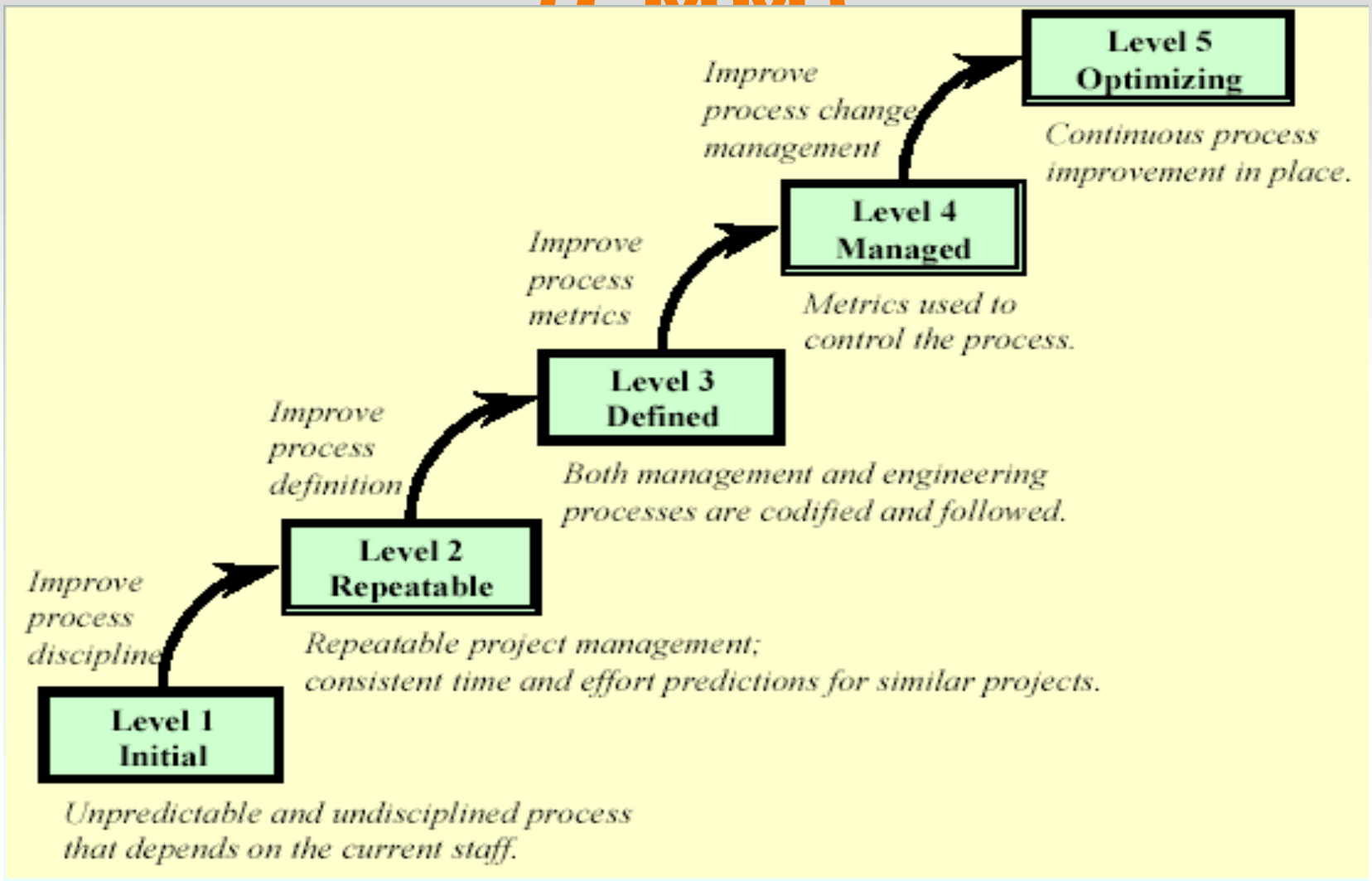
Questionario

Difficile salire di livello, facile scendere

Applicabilita’ a diversi settori ?

Necessario/sufficiente al successo? (Apple)

Capability Maturity Model (CMM)



CMM:requisiti livello 2

esistono procedure formali per:

tempi

rilevazione errori

statistiche

costi

controllo modifiche requisiti

controllo configurazione

controllo qualita’ indipendente

……

ISO 900*

Gestione della qualita’ nella costruzione e non alla fine

Prodotti di qualita’ da processo di qualita’

Lo standard specifica che cosa fare , non il processo (come)

Test: funzionare anche cambiando forza lavoro

Certificazione

- Documentare e registrare ogni attivita’
- Audit di quanto svolto

ISO 900*

Ditte certificate piu' profittevoli (causa o correlazione?)

Poco utile se richiesto solo dai clienti

The business:

- makes sure no one uses a bad product,
- determines what to do with a bad product,
- deals with the root cause of problems, and
- keeps records to use as a tool to improve the system.

Qualita' puo' avere molti aspetti

Pianificazione di sistema

Esempi di approcci:

- 5 forze di Porter
- SWOT
- VCM
- BPR
- ISA (Zachman)

Enfasi su efficacia (fare cosa giusta) piu' che su efficienza ,che sara' realizzata dal SW (si spera)

5 forze (minacce?) di Porter

Concorrenti diretti: soggetti che offrono la stessa tipologia di prodotto sul mercato

Fornitori: coloro dai quali l'azienda acquista materie prime e semilavorati necessari per svolgere il processo produttivo e che potrebbero decidere di integrarsi a valle;

Clienti: i destinatari dell'output prodotto dall'impresa che potrebbero eventualmente decidere di integrarsi a monte;

Potenziali entranti: soggetti che potrebbero entrare nel mercato in cui opera l'azienda;

Produttori di beni sostitutivi: soggetti che immettono sul mercato dei prodotti diversi da quelli dell'impresa di riferimento, ma che soddisfano, in modo diverso, lo stesso bisogno del cliente/consumatore.

Strengths,Weaknesses, Opportunities,Threats

Parte da missione aziendale

Punti interni di forza e debolezza

Opportunita’ e pericoli esterni

Obiettivi strategici (a lungo termine) concretizzati in obiettivi specifici misurabili

Politiche e tattiche (a breve termine) per la realizzazione concreta

Usato sempre relativamente ad un problema definito


Helpful
to achieving the objective
Harmful
to achieving the objective
Internal origin
(attributes of the organization)
Strengths
S
Weaknesses
W
External origin
(attributes of the environment)
Opportunities
O
Threats
T

Strengths,Weaknesses, Opportunities,Threats

Uso creativo: generazione di strategie

Trovare vantaggi competitivi col match di strengths and opportunities

Trasformare minacce e debolezze in forze ed opportunita'

Se un concetto in una analisi SWOT non genera strategie non e' importante

Political Economical Social Technological (PEST)

Approccio di analisi di un mercato

Prima di SWOT, che valuta un 'idea, od una impresa

E' come in mille altri casi una checklist

A che servono le checklist ?

Potete fare tabelle di pro e contro

In generale molto piu' a monte del nostro SI

Fattori PEST

Inquadrabili come opportunita' o minacce in analisi SWOT:

- Politici
- Economici
- Sociali
- Tecnologici
- Ambientali
- Giuridici
- Etici

PEST - political

ecological/environmental issues

current legislation home market

future legislation

european/international legislation

regulatory bodies and processes

government policies

government term and change

trading policies

funding, grants and initiatives

home market lobbying/pressure groups

international pressure groups

wars and conflict

PEST - economy

home economy situation

home economy trends

overseas economies and trends

general taxation issues

taxation specific to product/services

seasonality/weather issues

market and trade cycles

specific industry factors

market routes and distribution trends

customer/end-user drivers

interest and exchange rates

international trade/monetary issues

PEST - social

lifestyle trends

demographics

consumer attitudes and opinions

media views

law changes affecting social factors

brand, company, technology image

consumer buying patterns

fashion and role models

major events and influences

buying access and trends

ethnic/religious factors

advertising and publicity

ethical issues

PEST - technology

competing technology development

research funding

associated/dependent technologies

replacement technology/solutions

maturity of technology

manufacturing maturity and capacity

information and communications

consumer buying mechanisms/technology

technology legislation

innovation potential

technology access, licensing, patents

intellectual property issues

global communications

Value Chain Model (Porter,1985)

Dai materiali in ingresso al prodotto in uscita

Attivita' fondamentali:

- input,output,trasformazione,vendite,servizi

Attivita' di supporto:

- amministrazione e infrastruttura, personale, ricerca, SI

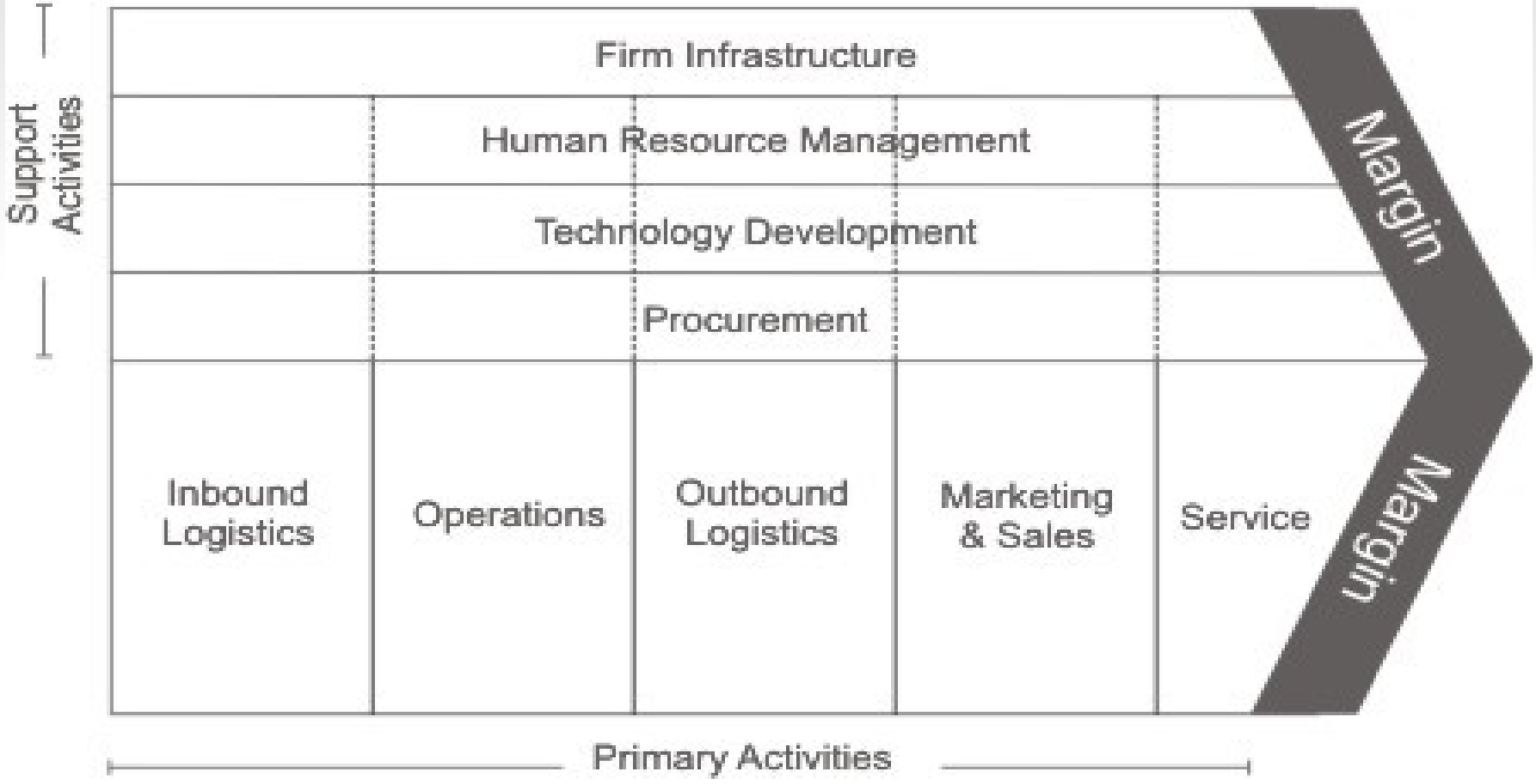
Un anello debole puo' spezzare la catena

Non modifica le unita' interne

Ha obiettivi strategici (a lungo termine)

Value Chain Model (Porter,1985)

Figure 1: Porter's Generic Value Chain



Value Chain Model ed IT

Stimare contenuto informativo prodotti e processi

identificare e graduare introduzione IT per creare vantaggio competitivo

creazione di nuovi business tramite IT

fare un piano per trarre vantaggio da IT

Business Process Reengineering

Prima:

- Organizzazione divisa verticalmente in Funzioni
- Nessun responsabile di processo
- Cambiamenti lenti e prevedibili

Adesso:

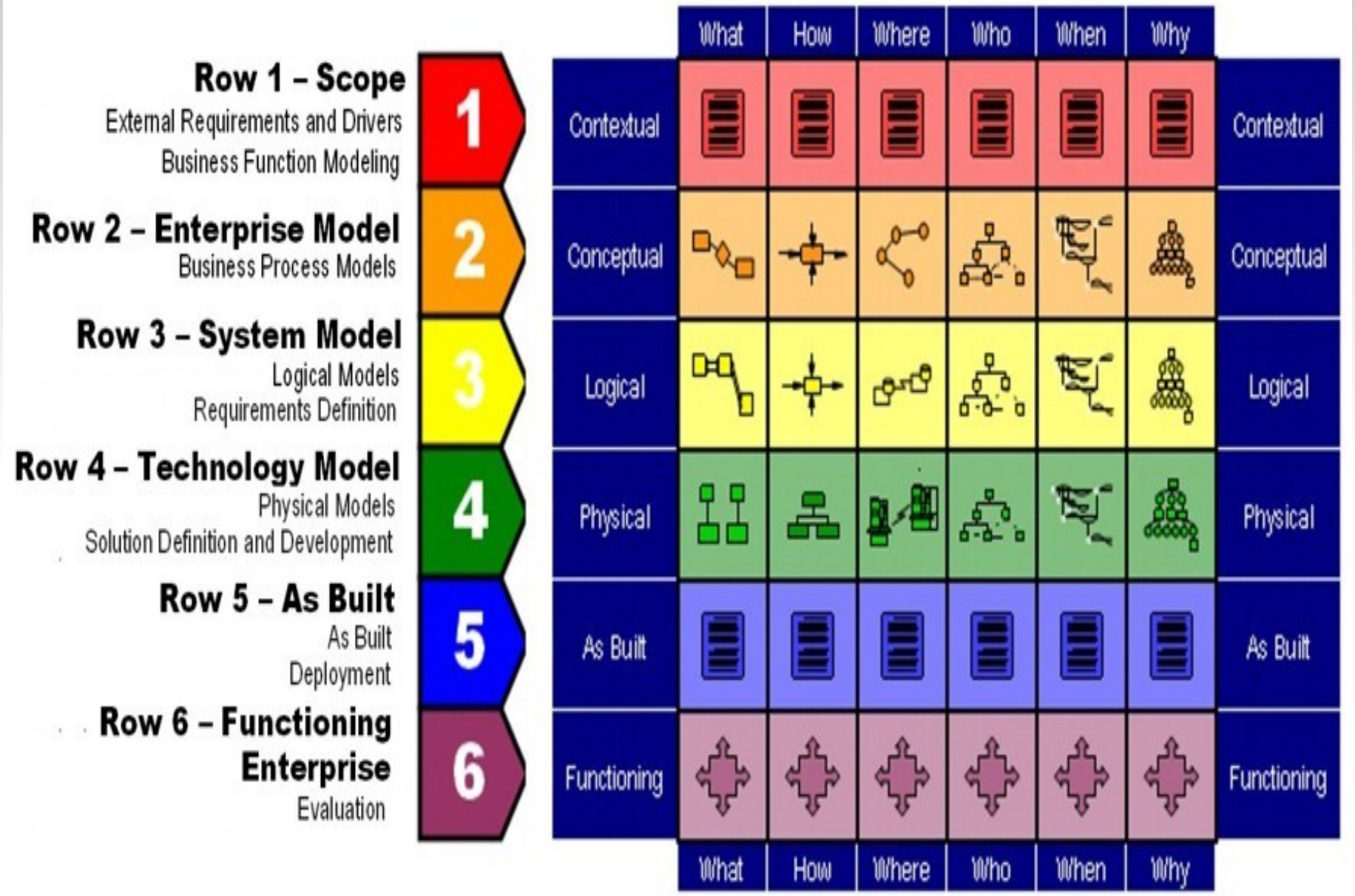
- Organizzazione a prodotti/ matrice
- Il processo attraversa orizzontalmente l'organizzazione fino al cliente
- Analisi del flusso di lavoro (workflow),per costi e benefici
- Ristrutturazione del processo ->
  - Difficolta' nel cambiare una organizzazione
- Miglioramento senza modifica radicale (BPImprovement) ->
  - Efficacia BPR + efficienza IT

ISA (Zachman, 1987)

Ontologia (schema descrittivo) flessibile dall'alto che supporta modelli di pianificazione a piu’ basso livello

Tabella con 36 celle

- Riconosce che la stessa cosa puo' essere descritta in modi differenti per scopi ed utenti differenti


Row 1 – Scope
External Requirements and Drivers
Business Function Modeling
Row 2 – Enterprise Model
Business Process Models
Row 3 – System Model
Logical Models
Requirements Definition
Row 4 – Technology Model
Physical Models
Solution Definition and Development
Row 5 – As Built
As Built
Deployment
Row 6 – Functioning Enterprise
Evaluation
1
2
3
4
5
6
What
How
Where
Who
When
Why
Contextual
Conceptual
Logical
Physical
As Built
Functioning

	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*
Objective/Scope (contextual) *Role: Planner*	List of things important in the business	List of Business Processes	List of Business Locations	List of important Organizations	List of Events	List of Business Goal & Strategies
Enterprise Model (conceptual) *Role: Owner*	Conceptual Data/ Object Model	Business Process Model	Business Logistics System	Work Flow Model	Master Schedule	Business Plan
System Model (logical) *Role:Designer*	Logical Data Model	System Architecture Model	Distributed Systems Architecture	Human Interface Architecture	Processing Structure	Business Rule Model
Technology Model (physical) *Role:Builder*	Physical Data/Class Model	Technology Design Model	Technology Architecture	Presentation Architecture	Control Structure	Rule Design
Detailed Reprentation (out of context) *Role: Programmer*	Data Definition	Program	Network Architecture	Security Architecture	Timing Definition	Rule Speculation
Functioning Enterprise *Role: User*	Usable Data	Working Function	Usable Network	Functioning Organization	Implemented Schedule	Working Strategy

Destinatari righe ISA

Pianificatore o investitore

Committente

Architetto

Costruttore

Appaltatore o componente acquisito

Impresa funzionante

Supporto IT/IS alla gestione (non come prodotto venduto)

Livello di management	Interesse	Applicazione IS	Soluzione IT
Strategico	Supporto ob. Lungo termine	Analisi vendite, pianif. prodotto	Data mining, knowledge management
Tattico	Allocazione ottimale risorse	Budget, gestione magazzino,logistica	Data warehouse, spreadsheets
Operativo	Attiv. giornaliere supporto al personale	Stipendi, acquisti, contabilita’	Basi dati, transazioni

Quando informatizzare ?

PRIMO: Facilitare il lavoro degli addetti

Proceduralita’ – lo svolgimento di una mansione e’ codificato esattamente (es contabilita’)

Alti volumi (banca) vs. studio professionale

Frequenza/ripetitivita’ (prenotazione aerea)

Riduzione errori

- es. consistenza/correttezza/duplicazione dati

Semplicita’ / standardizzazione

Quando informatizzare ?

Occorre tenere conto del modo di lavorare abituale di chi sara' l'utlizzatore del sistema

In generale, persone abituate a lavorare in modo disorganizzato potrebbero trovare difficile il salto in un modo molto proceduralizzato.

E' probabile quindi che l'introduzione di un SI sia piu' semplice in contesti dove gli addetti lavorano gia' in modo organizzato e metodico.

Linguaggi e strumenti di supporto

Linguaggi:

- Visuali (grazie alle Graphical User Iinterface)
- Dichiarativi: che cosa bisogna fare, non come (ne parliamo in seguito)

Ambiente integrato di Computer Aided Software Engineering (le macchine utensili del SW)

- Editing, building….
- Archivio (repository)
- Collaborazione/coordinazione
- Versioni/differenze
- Consistenza ed integrita’ dei modelli
- Generazione forward/backward e documentazione

Specifica(descrizione) mediante UML

Sviluppato da Rational Software Corporation

Approvato come standard da Object Management Group

Object Oriented (inadatto a implementazione procedurale)

Universale, indipendente dal tipo di SW descritto, dall' implementazione

Estensibile

Modella:

- Stati
- Comportamento
- Struttura

Vista d'insieme: il ciclo di vita del software

Ad alto livello:

- Analisi
- Progettazione
- Implementazione

La Pianificazione interseca tutto

In dettaglio (non strettamente sequenziali):

- Determinazione requisiti
- Specifica requisiti
- Progetto architetturale
- Progetto dettagliato
- Implementazione
- Integrazione
- Test

Determinazione dei requisiti

Requisito e' l' espressione di un servizio fornito dal sistema o di un vincolo cui deve sottostare

- Servizio:
  - Norma o consuetudine dell'organizzazione
  - Elaborazione di dati
- Vincoli: condizioni aggiuntive che limitano il campo delle soluzioni possibili

Determinazione dei requisiti

Attivita':

- Raccolta informazioni/negoziazione
- Prototipazione rapida
- Stesura del documento dei requisiti

Le precedenti attivita' sono eseguite in parallelo perche' si facilitano vicendevolmente

Specifica dei requisiti

Dal documento dei requisiti si passa a quello di specifica, cioe' descrizione dettagliata

Aggiunta requisiti prestazioni, manutenzione …

Diagrammi UML di visualizzazione

- Diagramma delle classi
- Diagramma dei casi d’uso

Obiettivi: generalita' e massima indipendenza dall’ implementazione hw/sw

Generalita’ -> massima flessibilita’ rispetto alla variazione dei vincoli (sempre utile ?)

Progetto dell'architettura

Soluzione a strati ormai standard per un IS:

- Client
- Server
- Strato di logica dell'applicazione

Strati composti a loro volta da componenti

Diagrammi UML:

- Package
- Componenti
- Deployment (installazione)

Progetto dettagliato

Interfaccia utente sul **client** (GUI)

Base dati sul **server**

Logica dell'applicazione -> Algoritmi

Diagrammi in UML:

- Classi
- Casi d'uso
- Attivita'
- Sequenza
- Collaborazione
- Stati

Implementazione

Installazione componenti acquisiti

Codifica

- Generazione scheletro codice da UML

Caricamento test e base dati di prova

Testing

Tuning (aggiustamento parametri di funzionamento)

Ottimizzazione

Addestramento utenti/manualistica

Implementazione

Sviluppo/Compilazione:

editor sintattico e compilatore supportano questa fase. Due sono gli aspetti piu' difficili:

- Permettere di eseguire tramite GUI tutte le operazioni permesse nel linguaggio relativo
- Mantenere consistenza tra l'ambiente sottostante e le informazioni relative alla GUI

Modificare la struttura del programma (refactoring), es cambiare nome a variabili/componenti etc.

Implementazione

Testing /Debugging:

Il debugger consente di ispezionare il codice in esecuzione e le strutture dati; piu' facile oggi coi linguaggi OO

Il testing presenta delle difficolta' quando si cerca riproducibilita' nei programmi con GUI

Implementazione

Profiling/Tuning

Un programma detto **profiler** misura il numero di chiamate ad un metodo e/o il tempo trascorso in esso producendo varie statistiche

In base ai dati si valuta se esistono pezzi di SW da riscrivere per aumentare l'efficienza

Tuning: se abbiamo previsto dei parametri di esecuzione, li variamo per migliorare le prestazioni

Implementazione

Documentazione

Attivita’ importantissima per rendere comprensibile il codice ai manutentori; molto semplificata e standardizzata tramite i tools dei vari IDE

Oggi per fortuna molta documentazione e' associata ai componenti standard che utilizzo.

Implementazione (5)

Controllo **versioni** importante sotto due aspetti:

Durante lo sviluppo, per conservare storia, versioni e varianti del programma

Per costruire versioni specifiche: “costruiamo una versione col componente X di dicembre e quello Y di gennaio”

Implementazione (6)

Costruzione ed assemblaggio (automatico) dei componenti del programma:

Ant/gradle/maven...

- legge file scritti in XML che specificano le dipendenze tra componenti
- Se un componente e' obsoleto, viene recuperata automaticamente la versione aggiornata dall' archivio su internet
- Per ogni componente esegue comandi di costruzione (compilazione, copiatura file, archiviazione ….)

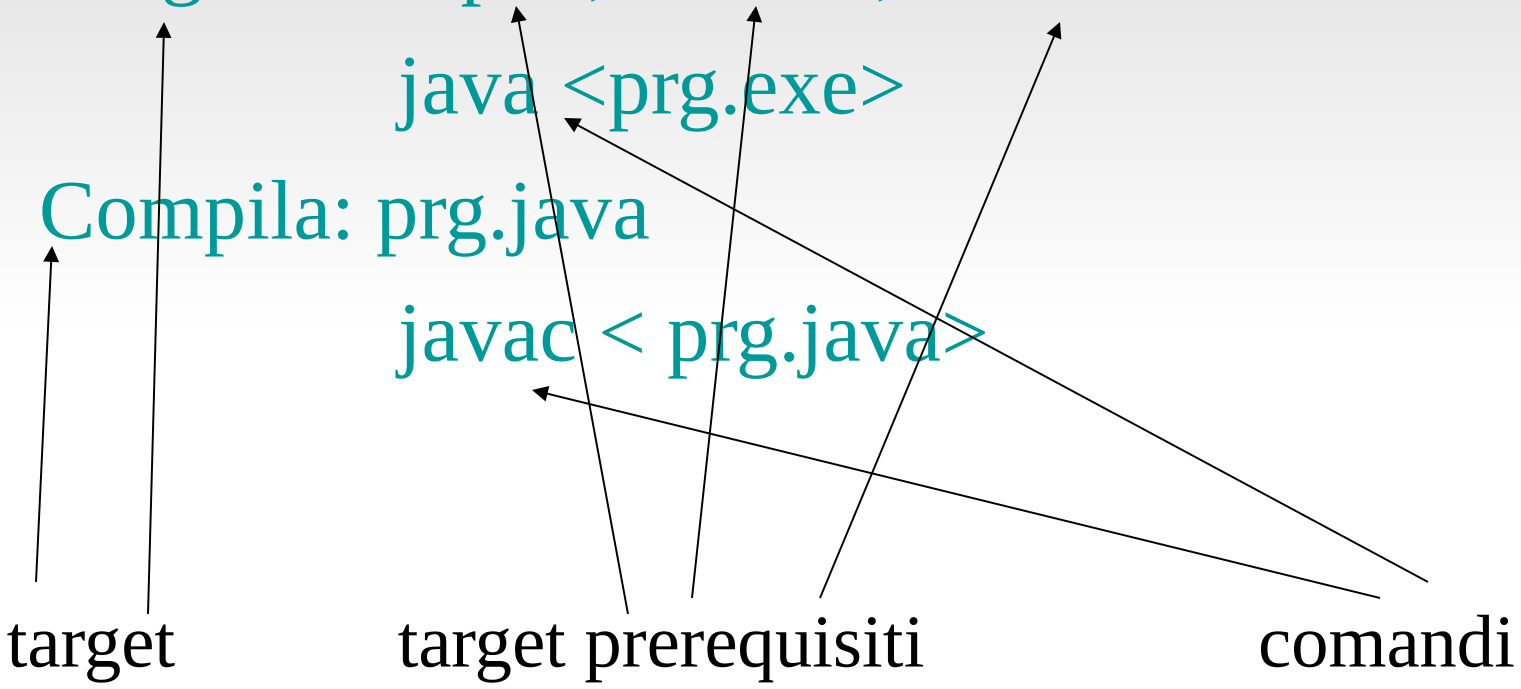
ant(linguaggio dichiarativo)

Esegui: compila, installa, crea archivio

java <prg.exe>

Compila: prg.java

javac < prg.java>



Creazione di un prodotto

IL SW a questo punto e' funzionante

Che cosa manca?

- Documentazione per l'utilizzatore
- Servizio assistenza cliente
- Configurazione prodotto
- Installazione prodotto
- Creazione delle release e delle patch

Cio' nell'ipotesi che le fasi precedenti siano corrette e che quindi il programma risponda ai requisiti di usabilita', prestazioni etc decise durante il progetto!

insieme di prodotti e versioni

a volte, il prodotto SW e' costituito da un insieme di componenti gestiti da team differenti.

Se, per ragioni commerciali le nuove versioni integrate devono uscire entro tempi prestabiliti, per evitare ritardi si utilizza un modello a treno:

- il treno parte, e man mano carica le prestazioni della nuova versione: se qualcuna non e' pronta viene abbandonata

Altro passo indietro: la pianificazione del progetto

Inizia dopo che il System Planning ha individuato quale Sistema Informativo realizzare.

Vincoli fissi:

- Tempo
- Denaro

Fattibilita' del progetto:

- Operazionale
- Economica
- Tecnica
- Temporale

Si producono due documenti: SPMP e SAD

Software Project Management Plan

Ambito del progetto

Attivita’

Direzione e controllo del progetto

Gestione della qualita’

Gestione delle misure

Organizzazione temporale (scheduling)

Allocazione risorse (persone , materiali, strumenti)

Gestione del personale

Software Architecture Document

Per introdurre nuovi partecipanti,

N+1 viste , 10 pagine

Contiene diverse “viste”

Documento di architettura software (SAD)

Vista logica

- Organizzazione concettuale, sottosistemi strati, componenti importanti
- Scenari piu' importanti dei casi d'uso

Vista processi

- Processi, thread, loro allocazione nei componenti

Documento di architettura software (SAD)

Vista di installazione

- Distribuzione dei componenti sull'HW

Vista dei dati

- Dati principali e quelli persistenti

Vista sicurezza

- Punti dove sicurezza ed autenticazione sono applicate

Documento di architettura software (SAD)

Vista di implementazione

- Sorgente ed eseguibili (I “deliverable”)

Vista di sviluppo

- Dove stanno I file, I repository, I test ..

Vista dei casi d'uso

- Quelli che mettono in luce gli aspetti piu' significativi del sistema

Vista dall'alto: **Evoluzione Software in prospettiva**

Fasi di una tecnologia: nascita, proliferazione, selezione, convergenza

Molti componenti software di funzionalita' sempre piu' complessa tendono alla standardizzazione (es.librerie Java)

Anche la configurazione del prodotto diventa standardizzata

Evoluzione Software

Il software e’ gia’ stato tutto scritto?

Il software sta conquistando il mondo (Web of Things)

Lo sforzo del programmatore diventa di integrazione di moduli e librerie:

Si scrive meno codice ma e’ piu’ difficile

Occorre apprendere moduli complessi

La tecnica e’ quella della configurazione

Proliferazione degli strumenti

Effimerita" delle competenze

Chi l’avrebbe detto

- **''Penso che ci sia posto, sul mercato mondiale, per circa 5 computer''**

  Thomas J. Watson, Amministratore Delegato IBM, 1948
- **''Che bisogno ha una persona di tenersi un computer in casa?''**

  Kenneth Olsen, fondatore della Digital Equipment, 1977
- **''640K saranno sempre sufficienti per chiunque''**

  Bill Gates, Microsoft, 1981

Riassumendo

Un IS e' un sistema sociale

Lo sviluppo software e' un processo artigianale creativo

Il successo richiede il coinvolgimento di tutti gli interessati, processo ,linguaggi e strumenti adeguati

Strategia di una organizzazione e pianificazione del sistema

Il ciclo di vita del sistema

Gli strumenti

I documenti

attivita’

Commenti / domande

Installazione netbeans bouml

Feedback! E Quiz su Moodle