

Progetto 1

Software Engineering

Valutazione comparativa di strumenti di ingegneria del software

Contenuti

- Modalità di svolgimento dei progetti
- Il progetto 1
- Contesto in cui eseguire la valutazione
- Tipologia di strumenti a supporto dell'Ingegneria del software
- Approccio metodologico alla valutazione
- Domande preliminari
- Aree di indagine
- Valutazione
- Rappresentazione dei risultati
- Discussione dei risultati



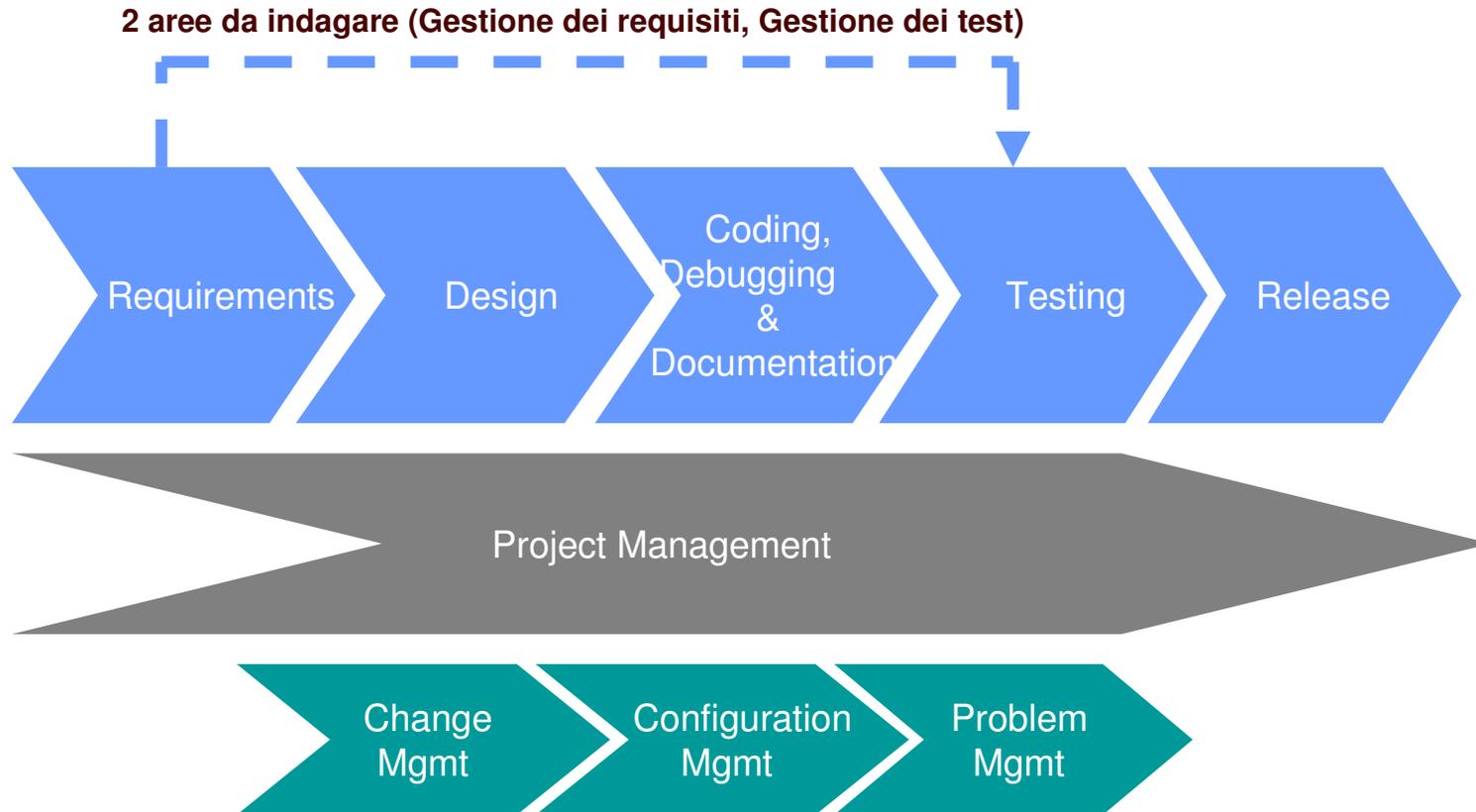
Modalità di svolgimento dei progetti

- Ogni progetto deve essere svolto da gruppi di minimo 2 persone fino ad un massimo di 4 persone (potrà essere svolto anche singolarmente dopo la terminazione del corso). Tutti i membri del gruppo riceveranno la stessa valutazione relativa al materiale prodotto ed un voto singolo relativo all'esposizione del progetto. Viene assunto che tutti i membri di ciascun gruppo abbiano contribuito fattivamente allo svolgimento dell'intero progetto.
- Saranno previste delle revisioni tecniche intermedie durante lo svolgimento del corso.
- La revisione tecnica finale del progetto sarà svolta alla fine del corso e prima della sessione di esame estiva.
- La consegna dei progetti e la relativa discussione sarà svolta durante la prova orale.

Il progetto 1: Valutazione comparativa di strumenti di ingegneria del software

- Il progetto consiste nell'analisi comparativa di 3 (tre) prodotti software (tool) presenti sul mercato per:
 - *Gestione dei requisiti software*
 - *Gestione dei test.*
- L'analisi comparativa riguarda le seguenti caratteristiche:
 - *Funzionalità (ricchezza funzionale offerta dal prodotto),*
 - *Flessibilità (capacità di personalizzare l'utilizzo delle funzioni offerte a seconda delle necessità dell'utente),*
 - *Integrazione del prodotto con altri (esempio, import ed export dei dati),*
 - *Portabilità (capacità di essere utilizzato su piattaforme diverse),*
 - *Costo del prodotto,*
 - *Altre caratteristiche che lo studente ritenesse interessanti.*
- La descrizione finale dell'analisi comparativa è presentata tramite
 - *un documento scritto con una struttura del tipo di quella mostrata alla fine della presentazione*
 - *aggiunta di grafici di tipo diverso che aiutino la comprensione.*

Contesto in cui eseguire la valutazione



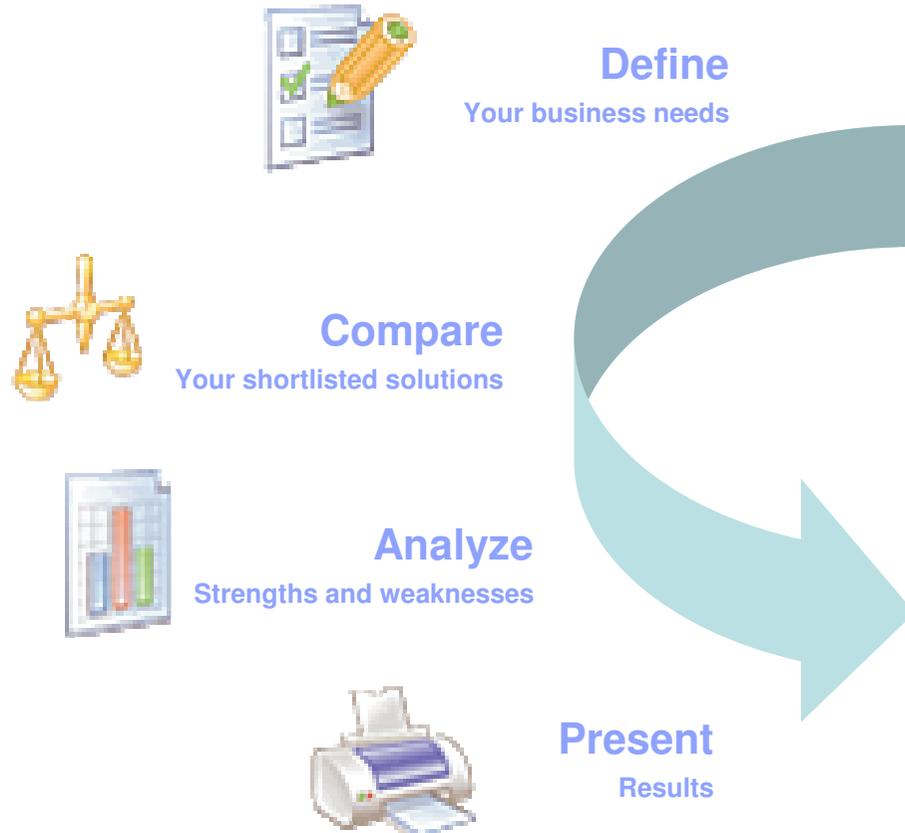
Tipologia di strumenti a supporto dell'Ingegneria del software

- **Software Lifecycle Management**
 - *Requirements Management*
- **Design and Analysis**
 - *Use Case Model*
- **Construction**
- **Documentation**
- **Verification & Validation**
 - *Verification (Technical Review)*
 - *Validation (Testing)*
- **Configuration Management**
- **Problem Management**

Approccio metodologico alla valutazione

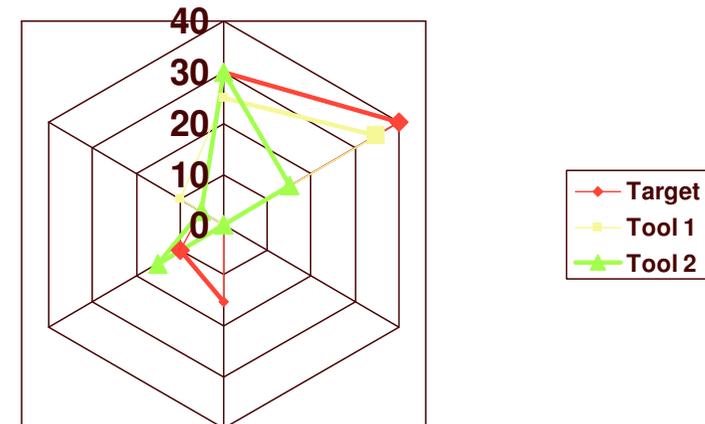
- Definire il contesto nel quale si pensa di utilizzare lo/gli strumento/i
- Definire gli obiettivi finali per cui si intende utilizzare lo/gli strumento/i
- Definire (in anticipo) i criteri di valutazione
- Effettuare la ricerca
- Registrare tutti i dati ricavati dalla ricerca
- Analizzare i dati secondo diversi punti di vista
- Valutare i risultati secondo i criteri definiti (a priori)
- Definire le diverse forme di rappresentazione dei risultati della ricerca
- Definire le diverse forme di rappresentazione dei risultati della valutazione
- Rappresentare i risultati della valutazione secondo le forme definite
- Rappresentare i dati di dettaglio secondo le forme definite
- Presentazione e discussione

Approccio completo



Caratteristica	Importanza
LifeCycle Management	40
Design and Analysis	10
Construction	0
Verification and Validation	40
Documentation	5
Configuration Management	0
Sw Dev Tool Technical Architecture	5
Total	100

Livello di aderenza al target



Domande preliminari

1. Obiettivi di business?
2. Funzionalità del tool?
3. Lingua supportata (italiano, inglese, ecc.)?
4. Piattaforma “target” del prodotto software realizzato?
5. Architettura della piattaforma “target”?
6. Linguaggio di programmazione?
7. Database?
8. Dimensione del sistema “target”?
9. Piattaforma del tool?
10. Standard?
11. Costi e scalabilità?
12. Settore d’industria?
13. Disponibilità geografica del tool e supporto?
14. Tempistica per l’implementazione?

1. Obiettivi di business

- Quali obiettivi di business deve supportare il tool?
 - Ridurre il tempo del ciclo di sviluppo (Lifecycle)
 - Ridurre i costi dello sviluppo
 - Aumentare la profittabilità del nuovo software
 - **Migliorare la qualità del software**
 - Migliorare i processi di revisione tecnica e di approvazione
 - Includere input degli utenti nel processo di design
 - Aumentare il livello di automazione del design
 - Aumentare il livello di riuso
 - Ridurre il tempo del ciclo di Change Management
 - Migliorare la collaborazione del team (locale e/o remoto)
 - Abilitare/Supportare l'esternalizzazione (Outsourcing) dello sviluppo
 - Costruire un Repository centrale per tutti i progetti
 - Altro

2. Funzionalità del tool

- Quali funzionalità principali deve offrire il tool?
 - **Gestione dei requisiti**
 - Gestione delle problematiche (issue)
 - Gestione delle attività (task) di progettazione e sviluppo
 - **Processo di modellazione del software** (se “Use Case” non incluso in “Gestione dei requisiti”)
 - Comprensione del software
 - Reverse engineering
 - Code editing
 - Codifica
 - **Verifica e validazione del software (testing)**
 - Rilascio del codice
 - Costruzione di interfacce grafiche
 - Costruzione di interfacce Web
 - Gestione delle versioni software
 - Gestione della costruzione del prodotto
 - Gestione del rilascio del prodotto software
 - **Creazione della documentazione del prodotto software**
 - Rilascio della documentazione del prodotto
 - API per l'integrazione con altri prodotti software
 - Altro

3. Lingua supportata dal tool

- Quali lingue deve supportare il tool?
 - Italiano
 - Inglese
 - Altre

4. Piattaforma “target”

- Su quale piattaforma “target” sarà installato il sistema software da realizzare?
 - IBM (Mainframe)
 - IBM (iSeries/AS400)
 - Linux
 - **Windows NT/2000/2003/XP** (è un esempio)
 - Windows 98/ME
 - Macintosh OS X
 - Unix
 - Unix-HP (HP-UX)
 - Unix-IBM (AIX)
 - Unix-Sun (Solaris)
 - Handheld device/Palmari (Linux, Palm OS, Windows CE)
 - Altri

5. Architettura della piattaforma “target”

- Quale architettura “target” dovrà supportare il sistema da realizzare?
 - Utente singolo (PC, Macintosh, palmare, Linux, Java, altro)?
 - Utente multiplo (Client/Server)?
 - Mainframe, batch, altro?
 - Web back-end software?
 - **Micorsoft .NET** (è un esempio)
 - J2EE?
 - Infrastruttura (sistema operativo, database, middleware, server, altro)?
 - Embedded?
 - Browser?
 - Altro?

6. Linguaggio di programmazione

- Quali linguaggi di programmazione si vogliono adoperare?
 - Ada
 - ASP
 - C, C++, C#
 - COBOL, Fortran, Pascal, PL/1, JCL
 - **HTML** (è un esempio)
 - **Java, JavaScript** (è un esempio)
 - JSP
 - Altro?

7. Database

- Quali Database “target” dovrà supportate il sistema da sviluppare?
 - **Oracle** (è un esempio)
 - IBM DB2
 - **Microsoft Access** (è un esempio)
 - Microsoft SQL Server
 - MySQL
 - Altro?

8. Dimensione del sistema “target”

- Quanti saranno i potenziali utenti finali del sistema che si intende sviluppare con il tool selezionato?
 - Utente singolo
 - da 2 a 10 utenti concorrenti
 - **da 11 a 100 utenti concorrenti**
 - da 101 a 500 utenti concorrenti
 - più di 500 utenti concorrenti

9. Piattaforma del tool

- In quale ambiente di sviluppo sarà installato il tool?
 - IBM iSeries (AS/400)
 - Linux
 - DEC Alpha
 - Macintosh OS X
 - Unix
 - **Windows NT/2000/XP**
 - Windows 98/ME
 - Unix (IBM AIX, HP-UX, Sun Solaris)
 - Altri

10. Standard

- A quali standard deve ottemperare il tool?
 - IEEE 1220
 - **ISO 9001**
 - **ISO/IEC 9126**
 - ISO/IEC 12119
 - ISO/IEC 12207
 - Altri (**es. UML**)

11. Budget e scalabilità

- Di quale budget disponete per l'introduzione del tool nella vostra organizzazione e quanti utenti (sviluppatori) dovranno utilizzarlo con accesso contemporaneo?

Numero di utenti

- Utente singolo
- **da 2 a 10 utenti**
- da 11 a 50 utenti
- più di 50 utenti

Budget

- fino a € 5.000,00
- € 10.000,00 - € 50.000,00**
- € 50.000,00 - € 200.000,00
- da definire

12. Settore d'industria “target”

• A quale settore d'industria è destinato il prodotto software che si intende sviluppare con il tool?

- Università
- Aerospaziale
- Banche
- Formazione
- Elettronica e HiTech
- Ingegneria e costruzioni
- Finanza
- Giochi
- Medicina
- Servizi interni
- Musica e film
- **Pubblica Amministrazione**
- Pubblicità
- Magazzini
- Sviluppo software
- Altri

13. Disponibilità del tool e supporto in loco

- Dove operate e dove vorreste fosse disponibile il supporto tecnico al tool?
 - Africa
 - Asia
 - Australia
 - Europa orientale
 - **Europa occidentale**
 - *Italia*
 - USA

14. Tempistica per l'implementazione

- Entro quanto tempo vorreste iniziare a sviluppare un nuovo progetto con il tool selezionato?
 - meno di 1 mese
 - da 1 a 3 mesi
 - **da 3 a 6 mesi**
 - da 6 mesi ad 1 anno
 - oltre 1 anno

Esempio di tool esistenti sul mercato

Alcuni prodotti esistenti sul mercato

- **Rational Suite (IBM)**
 - *RequisitePro (gestione dei requisiti)*
 - *TestManager (gestione dei test)*
- **ALM Suite (Borland)**
 - *CaliberRM (gestione dei requisiti)*
 - *Together (modellazione UML)*
 - *Optimizeit Suite e Optimizeit ServerTrace (gestione dei test di performance)*
- VisualStudio (Microsoft)
- ChangeMan – Application Lifecycle Automation (Serena)
- Tau (Telelogic)
- Objecteering (Softeam)
- Poseidon (Gentleware)
- Argo (Open source)
- UMLet (Open source)
- Popkin Software and Sybase
- Mercury Interactive (Mercury)

Nota: la lista mostrata non vuole assolutamente esprimere alcun giudizio di merito sui prodotti.

Definire quali caratteristiche interessa valutare

Area	Funzione	Sottofunzione	Elemento da valutare1.
1. LifeCycle Management	1. Requirements Management 2. Issue Management 3. Project Management 4. Standards Compliance		
2. Design and Analysis	1. Sw Process Modelling 2. Comprehension 3. Reverse Engineering 4. Migration		
3. Construction	1. Application 2. Database 3. User Interface		
4. Verification and Validation	1. V&V Management 2. V&V Data Architecture 3. V&V Automation		
5. Documentation	1. Content Creation 2. Content Methodology Support 3. Content Deployment		
6. Configuration Management	1. Version Management 2. Product Build Management 3. Release Management		
7. Sw Dev Tool Technical Architecture	1. Tool Execution Platform 2. Installation Architecture 3. Tool Data Architecture 4. Scalability		

Dettaglio: Requirements Management (1)

Area	Funzione	Sottofunzione	Elemento da valutare
1. LifeCycle Management	1. Requirments Management	1. Requirements Repository	<ol style="list-style-type: none"> 1. Unique requirements identification 2. Allow search of requirements 3. Allow the entry and update of requisrements 4. Requirements archiving 5. Enable attachment and storage of supporting documents 6. Support multiuser concurrency
		2. Requirements Traceability	<ol style="list-style-type: none"> 1. Track requirements creation date 2. Track requirements change date 3. Track resource(s) that create a requirement 4. Track resource(s) that updated a requirement 5. Associate a requirement with the software version 6. Ability to report on differences between versions of a requirement 6. Ability to search fo a requirement based upon key attributes 7. Ability to search for a requirement by text content
		3. Multiuser Collaboration	<i>Vedi pagina successiva</i>
		4. Integration	<i>Vedi pagina successiva</i>
		5. Customizability	<i>Vedi pagine successiva</i>

Dettaglio: Requirements Management (2)

Area	Funzione	Sottofunzione	Elemento da valutare
1. LifeCycle Management	1. Requirments Management	1. Requirements Repository	<i>Vedi pagina precedente</i>
		2. Requirements Traceability	<i>Vedi pagina precedente</i>
		3. Multiuser Collaboration	<ol style="list-style-type: none"> 1. Requirements change notification 2. Task management 3. On-line threaded discussion 4. Authorization and access mechanisms 5. Role-based authorization 6. User-based authorization 7. Record-level access restriction
		4. Integration	<ol style="list-style-type: none"> 1. Provides integration to project management and workflow tools 2. Provides integration to modelling tools 3. Provides integration to configuration management tools 4. Provides integration to testing tools .
		5. Customizability	<ol style="list-style-type: none"> 1. Customizable terms 2. Customizable categories 3. Enables definition of custom requirement attributes

Dettaglio: Design and Analysis

Area	Funzione	Sottofunzione	Elemento da valutare
2. Design and Analysis	1. Sw Process Modelling	1. Software Design Methodologies	1. Booch OOD; 2. Database ERD; 3. Data Flow Diagrams; 4. Jackson JSD; 5. Jacobson OOSE; 6. Rumbaugh OMT; 7. Yourdon & Coad OOA/OOD; 8. UML Activity; 9. UML Collaboration; 10. UML Component; 11. UML Deployment; 12. UML Sequence; 13. UML State Chart; 14. UML Static Structure; 15. UML Use Case
		2. Design Verification	1. Execute and verify data flow 2. Model UI flows 3. OMG MDA (Model Driven Architecture) standards conformance
		3. Code Generation	1. Business object generation 2. Business behaviour generation 3. Database DDL generation 4. Database persistence SQL
	2. Comprehension	Nessuna	1. Static analysis of the system; 2. Dynamic analysis of the system; 3. Call-tree diagram generation
	3. Reverse Engineering	Nessuna	1. Import source code into process model; 2. Import compiled code or byte-code into process model; 3. Maintain two-way synchronization of model and code
	4. Migration	Nessuna	1. Analysis of COBOL systems 2. Analysis of RPG systems 3. Analysis of PL/SQL systems

Dettaglio: Verification and Validation (Testing)

Area	Funzione	Sottofunzione	Elemento da valutare
4. Verification & Validation	1. Verification (Technical Reviews)	1. Prepare for verification	<ol style="list-style-type: none"> 1. Select Work Products for Verification 2. Assign Responsibilities 3. Establish Verification Criteria (severities, categories, types)
		2. Perform Peer Review	<ol style="list-style-type: none"> 1. Prepare for Peer Reviews (review documents) 2. Conduct Peer Reviews (store data) 3. Analyze Peer Review Data (stored data)
		3. Verify Selected Work Products	<ol style="list-style-type: none"> 1. Analyze Verification Results (produce reports) 2. Identify Corrective Action
	2. Validation (Testing)	1. Prepare for Validation	<ol style="list-style-type: none"> 1. Plan for Validation (Produce Test Plan) 2. Select Components for Validation 3. Define Validation Types 4. Define Validation Criteria (Entry and Exit)
		2. Design Validation	<ol style="list-style-type: none"> 1. Design Test Cases (input, activities, expected results) 2. Get Input from Use Cases (use UC to build Test Case) 3. Build/Update Requirements Traceability Matrix
		3. Validate Products and Components	<ol style="list-style-type: none"> 1. Execute Test Cases 2. Store Test Case Results 3. Interrupt/Resume Test Case Execution 4. Define Input Data to Test Cases (Oracle) 5. Manage Errors Status (Open, Assign, Resolve, Validate, Close), Data storage, and Reporting
		4. Verify Validation Results	<ol style="list-style-type: none"> 1. Verify Coverage Validation 2. Verify Validation Completion 3. Analyze Validation Results 4. Produce Reports

Interesse sulle caratteristiche da valutare

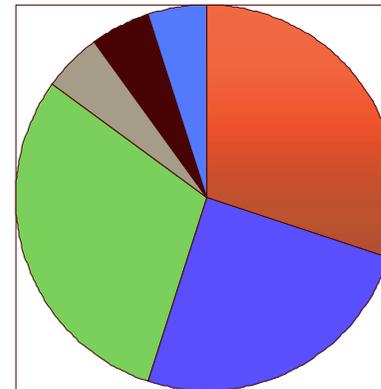
Area	Funzione	Sottofunzione	Elemento da valutare	Interesse
1. LifeCycle Management	1. Requirments Management	1. Requirements Repository	1. Unique requirements identification 2. Allow search of requirements 3. Allow the entry and update of requisrements 4. Requirements archiving 5. Enable attachment and storage of supporting documents 6. Support multiuser concurrency	Indispensabile Importante Indispensabile Indispensabile Poco importante Importante
		2. Requirements Traceability		
		3. Multiuser Collaboration		
		4. Integration		
		5. Customizability		
	2. Issue Management			

Interesse (per ciascun elemento da valutare):

Indispensabile	100%
Molto importante	80%
Importante	60%
Poco importante	20%
Non necessario	0%

Dimensionamento degli elementi da valutare

Elemento da valutare	Interesse
LifeCycle Management	Indispensabile
Design and Analysis	Importante
Construction	Non necessario
Verification and Validation	Indispensabile
Documentation	Poco importante
Configuration Management	Poco importante
Sw Development Tool Architecture	Poco importante



Interesse (per ciascun elemento; il totale deve risultare = 100%)

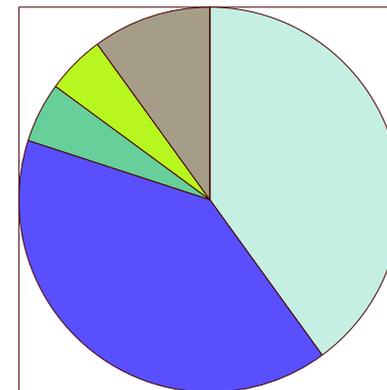
Indispensabile	100%
Molto importante	80%
Importante	60%
Poco importante	20%
Non necessario	0%

Analogamente per gli elementi di dettaglio

Elemento	Elemento		Interesse
LifeCycle Management	1. Requirements Management	1. Requirements Repository 2. Requirements Traceability 3. Multiuser Collaboration 4. Integration 5. Customizability	Indispensabile
	2. Issue Management		Non necessario
	3. Project Management		Non necessario
	4. Standards Compliance		Non necessario

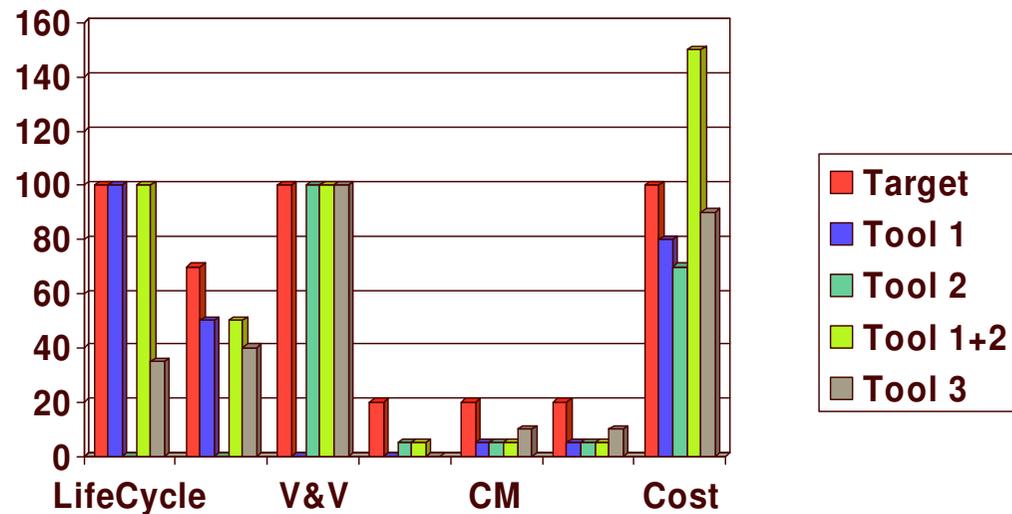
Interesse (per ciascun elemento)

Indispensabile	100%
Molto importante	80%
Importante	60%
Poco importante	20%
Non necessario	0%



Valutazione finale

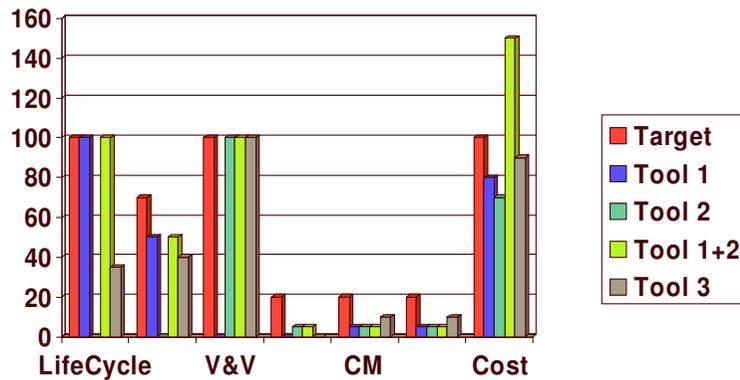
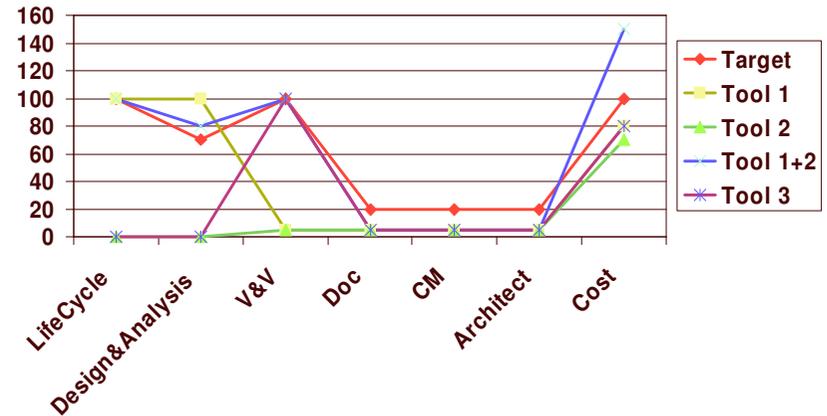
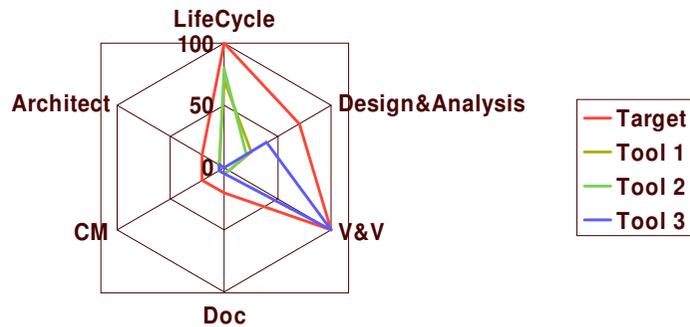
Caratteristica	Target	Tool 1	Tool 2	Tool 1+2	Tool 3
LifeCycle Management	100	100	0	100	35
Design and Analysis	70	50	0	50	40
Construction	0	0	0	0	0
Verification and Validation	100	0	100	100	100
Documentation	20	0	5	5	0
Configuration Management	20	5	5	5	10
Sw Dev Tool Technical Architecture	20	5	5	5	10
Cost	< €5000,00	€ 2500,00	€ 3500,00	€ 5000,00	€ 4500,00



Documentazione del lavoro eseguito

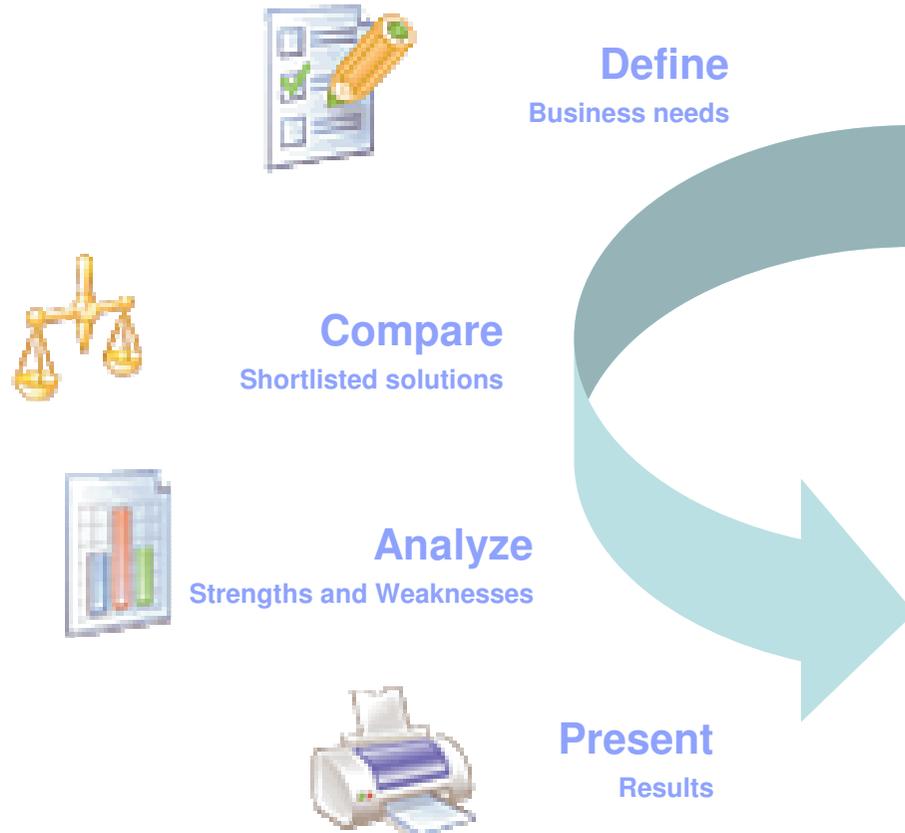
- Produrre un documento scritto con una struttura del tipo:
 1. Obiettivi del lavoro eseguito
 2. Metodologia adoperata
 3. Elementi di valutazione adoperati
 4. Scelta dei prodotti presi in considerazione
 5. Analisi delle caratteristiche valutate
 6. Valutazione delle caratteristiche
 7. Conclusioni finali
- La valutazione globale e quella delle singole caratteristiche può essere rappresentate tramite diverse tecniche:
 - Tabelle
 - Istogrammi, diagrammi polari, torte, diagrammi di Pareto, ecc.

Esempi di rappresentazione



Caratteristica	Target	Tool 1	Tool 2	Tool 1+2	Tool 3
LifeCycle anagement	100	100	0	100	35
Design and Analysis	70	50	0	50	40
Construction	0	0	0	0	0
Verification and Validation	100	0	100	100	100
Documentation	20	0	5	5	0
Configuration Management	20	5	5	5	10
Technical Architecture	20	5	5	5	10
Cost	<€5000,00	€ 2500,00	€ 3500,00	€ 5000,00	€ 4500,00

Approccio completo



Caratteristica	Importanza
LifeCycle Management	100
Design and Analysis	80
Construction	0
Verification and Validation	100
Documentation	5
Configuration Management	0
Sw Dev Tool Technical Architecture	5

Livello di aderenza al target

