



Università degli Studi di Udine

Manutenzione

prof. Maurizio Pighin

Dipartimento di Matematica e Informatica



Software maintenance

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Managing the processes of system change
- Modifying a program after it has been put into use
- Maintenance management is concerned with planning and predicting the process of change
- Configuration management is the management of products undergoing change

Slide 2





Program evolution dynamics

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Program evolution dynamics is the study of the processes of system change
- After major empirical study, Lehman and Belady proposed that there were a number of 'laws' which applied to all systems as they evolved
- There are sensible observations rather than laws. They are applicable to large systems developed by large organizations. Perhaps less applicable in other cases

Slide 3



Lehman's laws

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

Law	Description
Continuing change	A program that is used in a real-world environment necessarily must change or become progressively less useful in that environment.
Increasing complexity	As an evolving program changes, its structure tends to become more complex. Extra resources must be devoted to preserving and simplifying the structure.
Large program evolution	Program evolution is a self-regulating process. System attributes such as size, time between releases and the number of reported errors are approximately invariant for each system release.
Organisational stability	Over a program's lifetime, its rate of development is approximately constant and independent of the resources devoted to system development.
Conservation of familiarity	Over the lifetime of a system, the incremental change in each release is approximately constant.

Slide 4





Applicability of Lehman's laws

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- This has not yet been established
- They are generally applicable to large, tailored systems developed by large organizations
- It is not clear how they should be modified for
 - *Systems that incorporate a significant number of external components*
 - *Small organizations*
 - *Medium sized systems*

Slide 5



Maintenance is inevitable

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- The system requirements are likely to change while the system is being developed because the environment is changing. Therefore a delivered system won't meet its requirements!
- Systems are tightly coupled with their environment. When a system is installed in an environment it changes that environment and therefore changes the system requirements.
- Systems **MUST** be maintained therefore if they are to remain useful in an environment

Slide 6





Types of maintenance

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

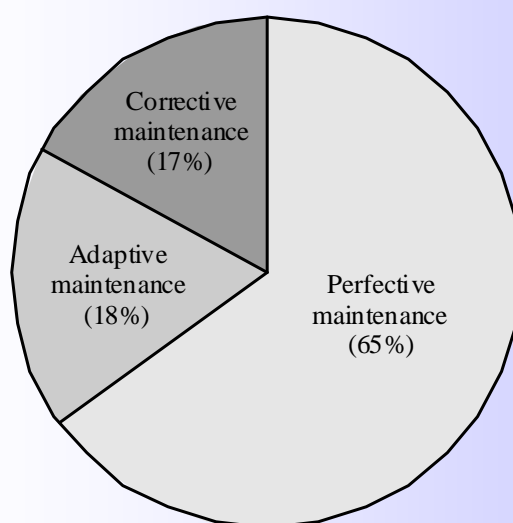
- Perfective maintenance
 - *Changing a system to make it meet its requirements more effectively*
- Adaptive maintenance
 - *Changing a system to meet new requirements*
- Corrective maintenance
 - *Changing a system to correct deficiencies in the way meets its requirements*

Slide 7



Distribution of maintenance effort

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo



Slide 8





Evolving systems

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

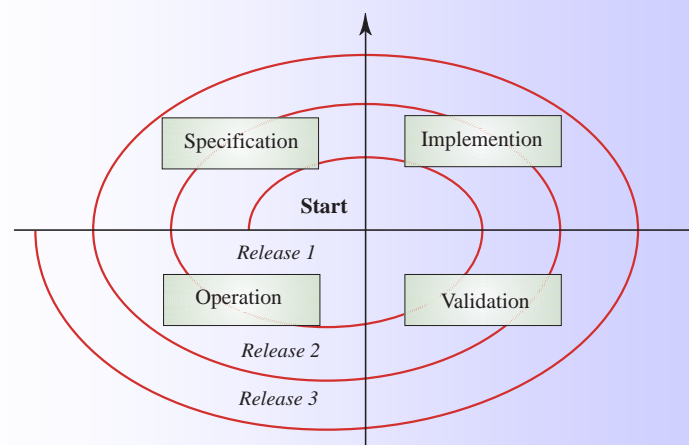
- It is usually more expensive to add functionality after a system has been developed rather than design this into the system
 - *Maintenance staff are often inexperienced and unfamiliar with the application domain*
 - *Programs may be poorly structured and hard to understand*
 - *Changes may introduce new faults as the complexity of the system makes impact assessment difficult*
 - *The structure may be degraded due to continual change*
 - *There may be no documentation available to describe the program*

Slide 9



Spiral maintenance model

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo



Slide 10





Maintenance management

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Maintenance has a poor image amongst development staff as it is not seen as challenging and creative
- Maintenance costs increase as the software is maintained
- The amount of software which has to be maintained increases with time
- Inadequate configuration management often means that the different representations of a system are out of step

Slide 11



Staff motivation

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Relate software development to organizational goals - maintenance rationale
- Relate rewards to organizational performance
- Integrate maintenance with development
- Create a discretionary preventative maintenance budget
- Plan for maintenance early in the development process
- Plan to expend effort on program maintainability

Slide 12





The maintenance process

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Maintenance is triggered by change requests from customers or marketing requirements
- Changes are normally batched and implemented in a new release of the system
- Programs sometimes need to be repaired without a complete process iteration but this is dangerous as it leads to documentation and programs getting out of step

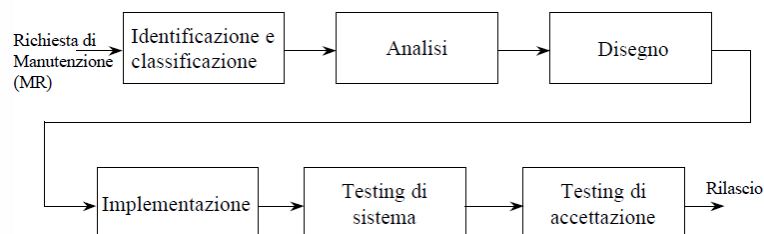
Slide 13



Lo standard IEEE 1291

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

Definisce un modello di processo di manutenzione di tipo iterativo articolato in sette fasi per gestire ed eseguire attività di manutenzione.



Slide 14





Identificazione e classificazione

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- In questa fase una richiesta di modifica (MR) è identificata, classificata, associata ad una priorità e valutata
- **Input**
 - *MR avanzata da: utenti, programmatori e/o manager*
- **Attività di processo della fase**
 - *codifica della MR e sua classificazione*
 - *analisi preliminare per decidere se accettarla, respingerla o valutarla ulteriormente*
 - *stima preliminare della sua dimensione/ampiezza*
 - *determinazione della sua priorità*
 - *assegnazione della MR a un blocco di modifiche già previste*

Slide 15



Identificazione e classificazione

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- **Attività di controllo della fase**
 - *memorizzazione della MR e dei risultati delle attività svolte in una base dati*
- **Output**
 - *I risultati delle attività svolte in termini di:*
 - *definizione del problema o dei nuovi requisiti*
 - *valutazione del problema o dei nuovi requisiti*
 - *classificazione della MR e sua priorità iniziale*
 - *verifica dei dati per le MR di tipo correttivo*
 - *stima iniziale delle risorse necessarie*

Slide 16





Analisi

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Vengono valutati fattibilità e scopo delle MR (analisi di fattibilità) e sviluppato un piano preliminare per il disegno, implementazione test e rilascio delle modifiche da apportare (analisi di dettaglio)
- **Input**
 - *l'output della fase precedente*
 - *la documentazione del sistema software (documentazione di sistema e di progetto)*

Slide 17



Analisi

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- **Attività di processo dalla fase**
 - *analisi di fattibilità per valutare:*
 - impatto
 - soluzioni alternative
 - analisi dei requisiti di conversione
 - fattori umani
 - costi a breve e a lungo termine
 - benefici ottenuti con le modifiche
 - *analisi dettagliata per:*
 - definire i requisiti richiesti all'azienda dalla modifica
 - individuare gli elementi da modificare
 - definire una strategia di testing
 - pianificare la realizzazione dell'intervento

Slide 18





Analisi

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- **Attività di controllo della fase**
 - *identificazione della versione più significativa della documentazione del sistema software*
 - *revisione delle modifiche proposte per validarne la fattibilità tecnico/economica e la correttezza*
 - *identificazione dei problemi di sicurezza*
 - *valutazione dell'integrazione delle modifiche nel software esistente*
 - *verifica che la documentazione disponibile è aggiornata verifica che esiste una strategia di testing e che questa è*
 - *idonea per le modifiche da apportare scelta delle modifiche da implementare nella nuova release*

Slide 19



Analisi

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- **Output**
 - *Report di:*
 - *fattibilità della MR*
 - *analisi dettagliata*
 - *requisiti da aggiornare*
 - *lista preliminare delle modifiche*
 - *strategia di testing da adottare*
 - *pianificazione delle attività di disegno ed implementazione*

Slide 20





Analisi

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Al termine della fase i risultati dell'analisi dettagliata dovranno consentire alla struttura aziendale preposta ad autorizzare le modifiche (struttura di gestione della configurazione) di decidere:
 - *se la richiesta debba essere soddisfatta o meno*
 - *in quale futura release dovranno essere inserite le modifiche*
- Naturalmente:
 - *gli interventi di manutenzione correttiva possono essere gestiti diversamente*
 - *deve essere opportunamente definita la procedura di rilascio di nuove release dovute ad interventi di manutenzione di emergenza*

Slide 21



Disegno (progetto)

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Vengono raccolte tutte le informazioni disponibili (documentazione del sistema software, risultati della fase di analisi, codice, ecc.)
- I documenti, peraltro già presi in esame durante la precedente fase di analisi, vengono rivisitati nell'ottica di progettare le modifiche.
- Le modifiche vengono progettate e sono aggiornati i documenti di analisi dettagliata

Slide 22





Disegno (progetto)

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- **Input**
 - *output della fase precedente*
 - *documentazione di sistema e di progetto*
 - *codice sorgente, commenti e basi di dati*
- **Attività di processo della fase**
 - *identificazione dei moduli affetti dalle modifiche*
 - *modifica della documentazione di disegno dei moduli software*
 - *definizione dei test case per le modifiche da apportare*
 - *identificazione/creazione dei test case di regressione*
 - *lista aggiornata delle modifiche*

Slide 23



Disegno (progetto)

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- **Attività di controllo della fase**
 - *ispezione dei documenti di disegno*
 - *verifica che i nuovi requisiti e il nuovo disegno è conforme a quanto autorizzato*
 - *verifica della inclusione della nuova documentazione in quella preesistente*
 - *verifica che la documentazione di testing è stata aggiornata*
 - *verifica della tracciabilità dai nuovi requisiti al nuovo progetto*

Slide 24





Disegno (progetto)

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- **Output**
 - *lista delle modifiche rivisitata*
 - *documentazione di disegno aggiornata*
 - *piani di test aggiornati*
 - *analisi di dettaglio aggiornata*
 - *requisiti verificati*
 - *piano dell'implementazione rivisitato*
 - *lista dei vincoli e dei rischi di fallimento*

Slide 25



Implementazione

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Le modifiche vengono effettuate sul codice.
- Viene inoltre effettuato:
 - *il testing di unità e di integrazione*
 - *l'aggiornamento di tutta la documentazione inclusi i documenti di disegno, di testing, la documentazione utente e di training*
- È altamente consigliabile che tutti gli oggetti costituenti il sistema software siano gestiti da un gestore automatico di configurazione che garantisce sia l'indipendenza delle varie versioni del sistema sia la consistenza di tutti gli oggetti afferenti alla stessa versione

Slide 26





Implementazione

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- **Input**
 - *output della fase precedente*
 - *documentazione di sistema e di progetto*
 - *codice sorgente, commenti e basi di dati*
- **Attività di processo della fase**
 - *modifiche sul codice*
 - *testing di unità*
 - *integrazione delle modifiche nel codice preesistente*
 - *testing di integrazione*
 - *rivisitazione dell'analisi dei rischi di fallimento*

Slide 27



Implementazione

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- **Attività di controllo della fase**
 - *ispezione del codice*
 - *verifica che il testing di unità e di integrazione sia stato fatto e documentato*
 - *verifica che piani di test, test case e procedure di testing siano stati aggiornati*
 - *verifica che il nuovo software sia stato posto sotto il controllo del gestore della configurazione software*
 - *verifica che la documentazione tecnica e di training è stata aggiornata*
 - *verifica della tracciabilità dal disegno al codice*

Slide 28





Implementazione

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- **Output**
 - *software aggiornato*
 - *documentazione di disegno aggiornata*
 - *piani di test aggiornati*
 - *documentazione utente aggiornata*
 - *documentazione di training aggiornata*

Slide 29



Testing di sistema

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Verifica che il sistema soddisfi i requisiti di cui alle modifiche e i requisiti originari.
- È uno dei punti critici del processo di manutenzione anche perché le teorie e le tecniche di testing sono orientate al testing per lo sviluppo piuttosto che verso il testing in manutenzione
- Richiede l'esecuzione del testing di regressione che è un testing specifico della manutenzione. A tal fine è essenziale che sia disponibile un database di test case che qualora manchi deve essere necessariamente creato e tenuto aggiornato

Slide 30





Testing di sistema

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- **Input**
 - *output della fase precedente*
 - *documentazione di testing di sistema (piani di test, test case, procedure di test)*
 - *manuali utente documentazione di disegno*
- **Attività di processo della fase**
 - *test funzionale di sistema*
 - *testing di interfaccia*
 - *testing di regressione*
 - *preparazione al testing di accettazione*

Slide 31



Testing di sistema

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- **Attività di controllo della fase**
 - *Sarebbe opportuno che le attività di testing fossero condotte da una “terza parte”.*
 - *Prima che la fase termini è necessario definire lo status, cioè stabilire la misura secondo cui i criteri di testing sono stati soddisfatti.*
- **Output**
 - *sistema testato e completamente integrato;*
 - *test report;*
 - *review dei test report.*

Slide 32





Testing di accettazione

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- È il classico testing di accettazione sviluppato dal cliente, utente o da una terza parte.
- Consiste nell'analisi di report di risultati di test e nell'audit delle funzionalità del sistema di cui si accetta la nuova release.
- Particolare cura deve essere posta in questa fase nel verificare che tutta la documentazione, sia di sistema che di utente, sia stata correttamente aggiornata.
- Un sistema la cui documentazione non è aggiornata non deve superare la fase di testing di accettazione e quindi non è pronto per la messa in esercizio

Slide 33



Rilascio

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- È la fase in cui il fornitore installa presso il cliente la nuova versione e la messa in esercizio
- Sono da effettuare in questa fase:
 - *una puntuale verifica della consegna di tutti gli oggetti*
 - *costituenti la nuova versione*
 - *il back-up della vecchia versione per archiviazione*
 - *Il training dell'utente sulla nuova versione rilasciata*

Slide 34





Manutenzione “d’urgenza”

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- In alcuni casi le richieste di manutenzione devono essere soddisfatte rapidamente:
 - *Se un difetto serio deve essere riparato;*
 - *Se modifiche dell’ambiente operativo causano effetti collaterali imprevisti sull’operatività del sistema;*
 - *Se è necessario l’adeguamento urgente a seguito di cambiamenti imprevisti (es. Ambiente o mercato).*
- In questi casi, le fasi di analisi e progetto della modifica potrebbero non essere eseguite, implementando direttamente il cambiamento.

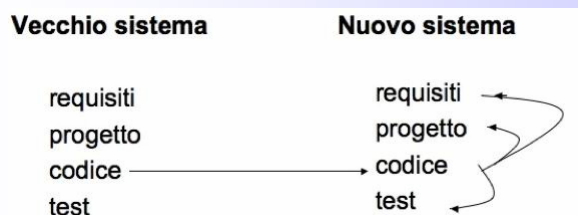
Slide 35



Manutenzione “d’urgenza”

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- **Processo di riparazione d’urgenza (quick- fix model)**
- Con tale approccio, la qualità complessiva del software si ridurrà.
- Utile pianificare post-rilascio interventi di aggiornamento della documentazione e/o di miglioramento della qualità del software.



Slide 36





Maintenance costs

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

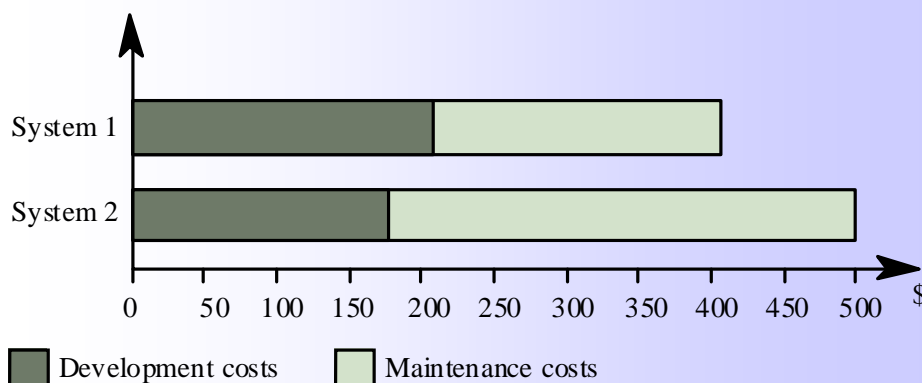
- Usually greater than development costs (2* to 100* depending on the application)
- Affected by both technical and non-technical factors
- Increases as software is maintained. Maintenance corrupts the software structure so makes further maintenance more difficult.
- Ageing software can have high support costs (e.g. old languages, compilers etc.)

Slide 37



Development/maintenance costs

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo



Slide 38





Maintenance cost factors

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Module independence
 - *It should be possible to change one module without affecting others*
- Programming language
 - *High-level language programs are easier to maintain*
- Programming style
 - *Well-structured programs are easier to maintain*
- Program validation and testing
 - *Well-validated programs tend to require fewer changes due to corrective maintenance*

Slide 39



Maintenance cost factors

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Documentation
 - *Good documentation makes programs easier to understand*
- Configuration management
 - *Good CM means that links between programs and their documentation are maintained*
- Application domain
 - *Maintenance is easier in mature and well-understood application domains*
- Staff stability
 - *Maintenance costs are reduced if the same staff are involved with them for some time*

Slide 40





Maintenance cost factors

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Program age
 - *The older the program, the more expensive it is to maintain (usually)*
- External environment
 - *If a program is dependent on its external environment, it may have to be changed to reflect environmental changes*
- Hardware stability
 - *Programs designed for stable hardware will not require to change as the hardware changes*

Slide 41



Maintenance cost estimation

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Can use COCOMO model for maintenance cost estimation
- Based on % of program instructions changed per year (ACT=Annual Change Traffic)
- Simplistically, maintenance costs are directly proportional to development costs
 $AME = ACT * STD$
 - *AME=Annual Maintenance Effort*
 - *STD=Software Developing Time*
- SW 236 Person/moth, 15% modified/year
 $AME = 0.15 * 236 = 35.4$ Person/month
- Multipliers like development multipliers but with different values

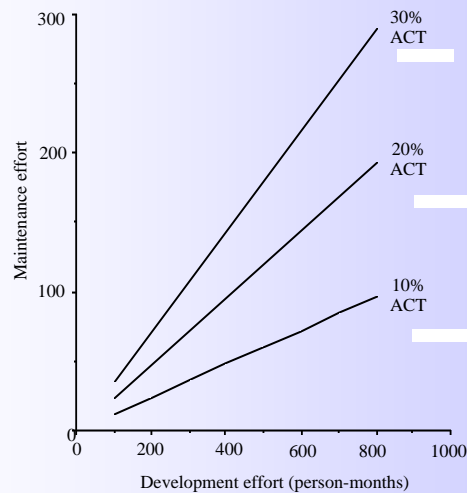
Slide 42





Maintenance/development effort

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo



Slide 43



Maintenance prediction

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Maintenance prediction is concerned with assessing which parts of the system may cause problems and have high maintenance costs
 - *Change acceptance depends on the maintainability of the components affected by the change*
 - *Implementing changes degrades the system and reduces its maintainability*
 - *Maintenance costs depend on the number of changes and costs of change depend on maintainability*

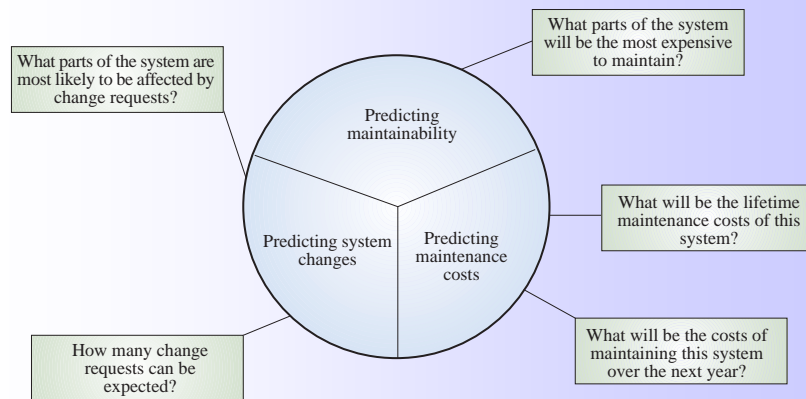
Slide 44





Maintenance prediction

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo



Slide 45



Change prediction

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Predicting the number of changes requires and understanding of the relationships between a system and its environment
- Tightly coupled systems require changes whenever the environment is changed
- Factors influencing this relationship are
 - *Number and complexity of system interfaces*
 - *Number of inherently volatile system requirements*
 - *The business processes where the system is used*

Slide 46





Maintenance prediction

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- COCOMO approach is simplistic because of non-technical factors
- COCOMO approach can be used to decide what and what kind of resources should be devoted to maintenance

Slide 47



Complexity metrics

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Predictions of maintainability can be made by assessing the complexity of system components
- Studies have shown that most maintenance effort is spent on a relatively small number of system components
- Complexity depends on
 - *Complexity of control structures*
 - *Complexity of data structures*
 - *Procedure and module size*

Slide 48





Process metrics

Ingegneria del software
Progettazione e Laboratorio
Manutenzione
Maurizio Pighin, Andrea Baruzzo

- Process measurements may be used to assess maintainability
 - *Number of requests for corrective maintenance*
 - *Average time required for impact analysis*
 - *Average time taken to implement a change request*
 - *Number of outstanding change requests*
- If any or all of these is increasing, this may indicate a decline in maintainability

Slide 49

