

Università degli Studi del Sannio – Facoltà di Ingegneria

MODELLI DI CVS

UN MODELLO DI CICLO DI VITA DEL SOFTWARE E' UNA CARATTERIZZAZIONE DESCRITTIVA O PRESCRITTIVA DI COME UN SISTEMA SOFTWARE VIENE O DOVREBBE ESSERE SVILUPPATO

Encyclopedia of Software Engineering
w.Scacchi, pag 861, Vol. II

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

1

Università degli Studi del Sannio – Facoltà di Ingegneria

	n.ro unità di prog	tempo, manp	dimensione
TRIVIAL	1	1 day-1 week 1	
SMALL	25-30	1-6 mesi 1	1000-2000 LOC
MEDIUM	250-1000	1-2 anni 2-5	5000-10000 LOC
LARGE	più sottosistemi	2-3 anni 5-20	50000-100000 LOC
VERY LARGE	più sottosistemi alcuni sono large	4-5 anni 50-200	1-2 milioni LOC
EXTREMELY LARGE	Sottosistemi very large	Più di 5 anni 200- 1000	> > 2 milioni LOC

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

2

Università degli Studi del Sannio – Facoltà di Ingegneria

- ❑ una morfologia del lavoro nelle fabbriche del software
 - fasce della produzione
 - tipi di attività
 - collegamento ed interfacciamento
 - controllo e misura
- ❑ linee guida per Organizzare, Pianificare, Dimensionare Personale, Assegnare Budget, Schedulare, Gestire

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

3

Università degli Studi del Sannio – Facoltà di Ingegneria

... definire e prescrivere prodotti e documenti da rilasciare al committente (deliverables)

... determinare, classificare e prescrivere metodi, tecniche e tecnologie della IS adatte a supportare le attività previste

.....

... apprendere, analizzare, migliorare

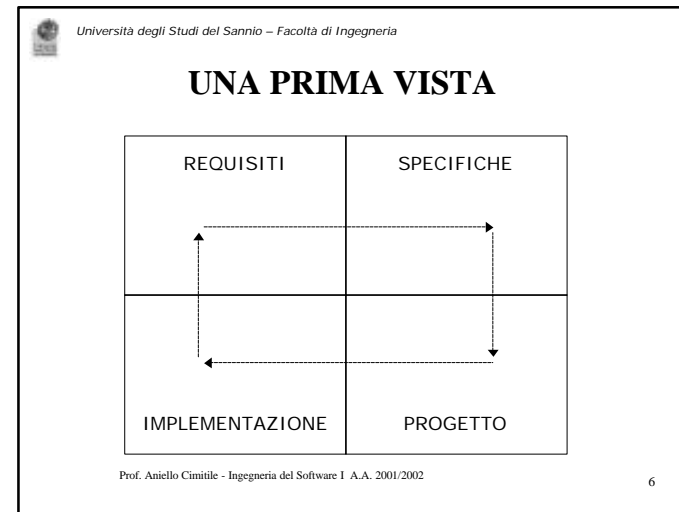
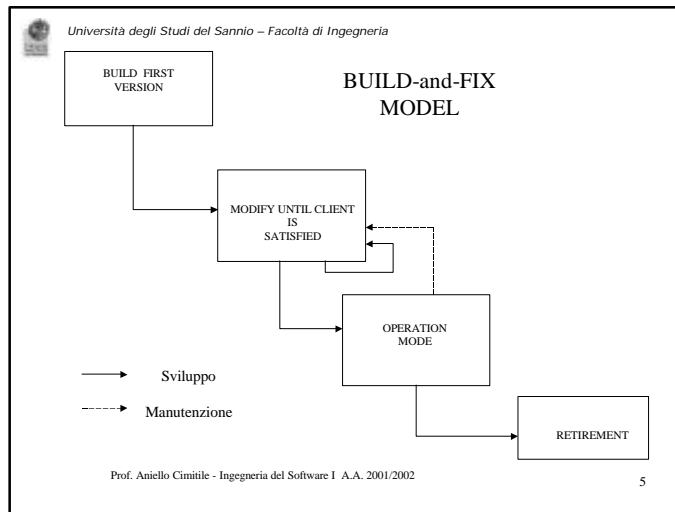
NON CONFONDERE con Modelli di Processo Software

precise e formalizzate descrizioni di dettaglio delle attività, degli oggetti, degli eventi, delle trasformazioni, etc. , che caratterizzano l'evoluzione del processo

Anche con l'uso di metodi formali (ad es. Reti di Petri)

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

4



Università degli Studi del Sannio – Facoltà di Ingegneria

MODELLI A CASCATA (WATERFALL MODELS)

Royce 1970
Boehm, McClure, Fairley, ...

SONO MODELLI SEQUENZIALI
... rigide conseguenze organizzative ...

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

7

- Università degli Studi del Sannio – Facoltà di Ingegneria
- ❑ Il ciclo è organizzato in FASI (stages or phases)
Ogni fase raccoglie un insieme di attività omogenee per metodi, tecniche, tecnologie, skill del personale, etc.
 - ❑ Ogni fase è caratterizzata da:
ATTIVITA' (tasks)
PRODOTTI (deliverables)
CONTROLLI E MISURE (V&V, Qualità)
 - ❑ La fine di ogni fase è punto rilevante del processo (milestone)
- Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002
- 8

Università degli Studi del Sannio – Facoltà di Ingegneria

- ❑ I deliverables di una fase (anche output della fase), sono input della fase successiva
- ❑ I prodotti di una fase vengono “congelati”, ovvero non sono più modificabili se non innescando un processo formale e sistematico di modifica

... è un punto critico ...

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

9

Università degli Studi del Sannio – Facoltà di Ingegneria

UN ESEMPIO DI WM

```

    graph TD
      A[ANALISI DEI REQUISITI E SPECIFICHE] --> B[PROGETTO]
      B --> C[CODIFICA]
      C --> D[TESTING]
      D --> E[MANUTENZIONE]
    
```

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

10

Università degli Studi del Sannio – Facoltà di Ingegneria

IL CICLO DI BOEHRM

VAL: validation
VER: verification

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

11

Università degli Studi del Sannio – Facoltà di Ingegneria

Zelkowitz '79
Gilb '83

PROBLEM DESIGN

↓

REQUIREMENT ANALYSIS

↓

SPECIFICATION

↓

DESIGN

↓

CODING

↓

TESTING

↓

OPERATION AND MAINTENANCE

Ghezzi et al. '91

STUDIO DI FATTIBILITA'

↓

ANALISI E SPECIFICA DEI REQUISITI

↓

PROGETTAZIONE

↓

PROGRAMMAZIONE E TEST DI UNITA'

↓

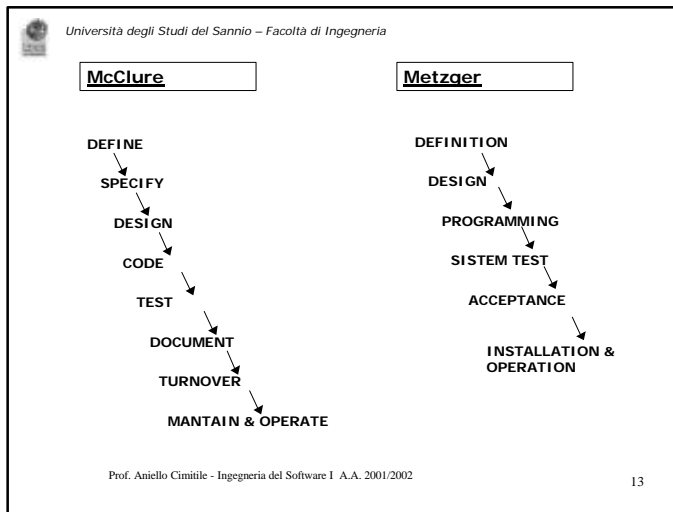
INTEGRAZIONE E TEST DI SISTEMA

↓

MANUTENZIONE

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

12



Università degli Studi del Sannio – Facoltà di Ingegneria

A. Von Mayrhauser '90

<u>FASI</u>	<u>DELIVERABLES</u>
PROBLEM DEFINITION	Documento di definizione del problema, proposta ed obiettivi, principali vincoli e prescrizioni
REQUIREMENT ANALYSIS	Documento dei Requisiti, budget preliminare, piano di massima, ..., criteri per testing di accettazione, studio di fattibilità
SPECIFICATION	Documento delle Specifiche, manuale di utente preliminare, test cases funzionali, piani di sviluppo
DESIGN	Documento di HLD, Documento di LLD, Test cases di progetto
CODE	Codice, Report Testing di Unità, Test Cases usati, verifica di standard
TESTING	Codice testato, report di testing, documenti finali (training, user, system), valutazione finale progetto
MAINTENANCE	Problem log, controllo versioni, documento aggiornato, documenti di cms

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002 14

Università degli Studi del Sannio – Facoltà di Ingegneria

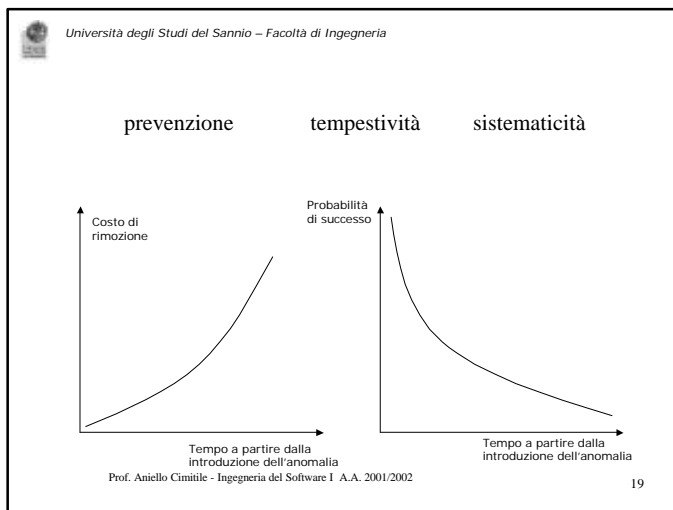
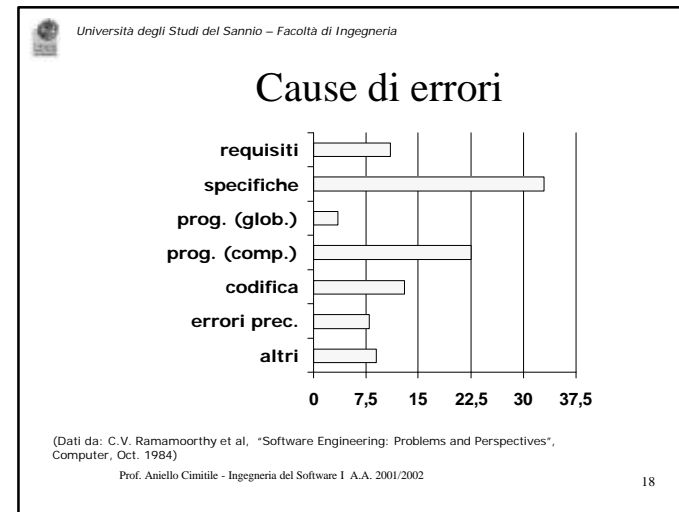
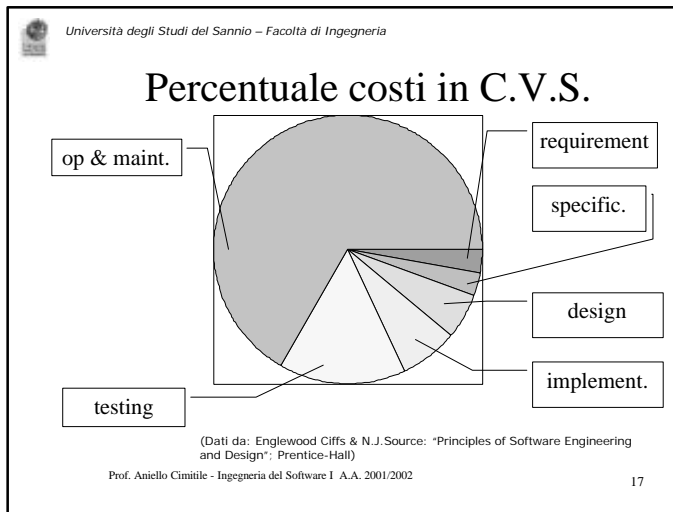
Quale WM ?

... molti aspetti influenzano la definizione del modello di CVS

- Specificità dell'organizzazione produttrice
- Know-how
- Dominio applicativo e particolare progetto
- Strumenti di supporto
- Risorse disponibili

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002 15

- Università degli Studi del Sannio – Facoltà di Ingegneria
- produttore e committente coincidono
 - produttore e committente sono entità organizzativamente distinte all'interno della stessa azienda o istituzione
 - il produttore sviluppa una specifica applicazione (... il pacchetto ...) da immettere sul mercato e non per uso utente
 - il produttore è anche erogatore dei servizi di gestione operativa
- Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002 16



Università degli Studi del Sannio – Facoltà di Ingegneria

FP	Early	On Time	Delayed	Canceled
1	14.68%	83.16%	1.92 %	0.25 %
100	6.06 %	74.77 %	11.83 %	7.23 %
10000	0.14 %	28.03 %	23.83 %	48.00 %
100000	0.00	13.67 %	21.33 %	65.00 %
MEDIA		56.94 %	13.71 %	23.82 %

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

20

Università degli Studi del Sannio – Facoltà di Ingegneria

Typical Work Year Breakdown

C. Jones 1998

Testing and defect repairs	70	35.53 %
Y2K and related repairs	50	25.38 %
Time on canceled project	30	15.23 %
Productive time on subprojects	47	23.86 %

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

21

Università degli Studi del Sannio – Facoltà di Ingegneria

Alcune considerazioni

- ❑ Distribuzione dei costi per fase
- ❑ Distribuzione cause di errore per fase
- ❑ Costo rimozione errore e probabilità di successo

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

22

Università degli Studi del Sannio – Facoltà di Ingegneria

Problemi

- ❑ Il nuovo sistema software diventa installabile solo quando è totalmente finito
- ❑ Né l'utente finale, né il management possono giudicare dell'adesione del sistema alle proprie aspettative prima della fine
- ❑ I "cambiamenti" non sono visti bene
- ❑ Impossibilità di overlap di fasi diverse

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

23

Università degli Studi del Sannio – Facoltà di Ingegneria

Modelli "DO IT TWICE"

- ❑ Si fondano su una prima implementazione che viene vista come una prova ed i cui obiettivi sono
 - Accertare la fattibilità del prodotto
 - Verificare i requisiti
- ❑ La seconda e definitiva versione del sistema software viene poi sviluppata con un modello di tipo WM

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

24

Università degli Studi del Sannio – Facoltà di Ingegneria

Modello con pilota

- ❑Viene costruito un sistema PILOTA per stabilire se nei suoi aspetti fondamentali il sistema software è fattibile... e dimostrando ciò attraverso l'implementazione
- ❑Il pilota è un sistema software completo, anche se "rough around the edges" è esplorativo non è orientato alla produzione
- ❑Un sistema pilota fornisce feedbacks ed indicazioni per la definizione di un prodotto migliore e più utile

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

25

Università degli Studi del Sannio – Facoltà di Ingegneria

Modelli con prototipo

- ❑Viene costruito un prototipo del sistema software in generale in modo veloce (rapid prototyping)
 - Per comprendere e definire dettagliatamente i requisiti di utente
 - Per comprendere subito se e come il sistema migliora la produttività ed è utile
 - Per implementare le funzionalità critiche del sistema valutandone subito le soluzioni
- ❑Una volta che il prototipo ha fornito tutti i feedbacks sulla esatta definizione dei requisiti di utente, su decisione dell'Ing. del Software, viene buttato (THROWAWAY PROTOTYPE) e viene avviato un regolare processo di produzione

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

26

Università degli Studi del Sannio – Facoltà di Ingegneria

Esempio di DIT2 con prototipo

```
graph TD; R[Requisiti] --> P[Progettazione veloce]; P --> PR[Prototipo]; PR --> VR[Valutazione e raffinamento requisiti]; VR --> R; VR --> CP[CVS prodotto];
```

Il diagramma illustra il ciclo di vita DIT2 con prototipo. Il processo inizia con i Requisiti, che portano alla Progettazione veloce. Questa si evolve in un Prototipo, che viene poi sottoposto a Valutazione e raffinamento requisiti. Questo passaggio di valutazione ha un feedback diretto sui Requisiti, permettendo iterazioni. Infine, il processo si conclude con il CVS prodotto.

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

27

Università degli Studi del Sannio – Facoltà di Ingegneria

Tipi di Prototipazione

- ❑MOCK-UPS: costruzione completa delle interfacce di utente l'analisi completa e l'uso delle interfacce da parte dell'utente consente di mettere a punto nel dettaglio come l'utente interagirà col sistema ed in ultima istanza consentirà di definire con completezza e senza ambiguità i requisiti

inoltre l'utente e l'Ing. del Software possono subito valutare utilità e produttività ...

- ❑BREAD BOARDS: implementazione del sottoinsieme delle funzionalità critiche del sistema, senza le interfacce di utente

esperienza e feedbacks su come implementare le funzionalità (...conoscere prima di garantire...)

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

28

Università degli Studi del Sannio – Facoltà di Ingegneria

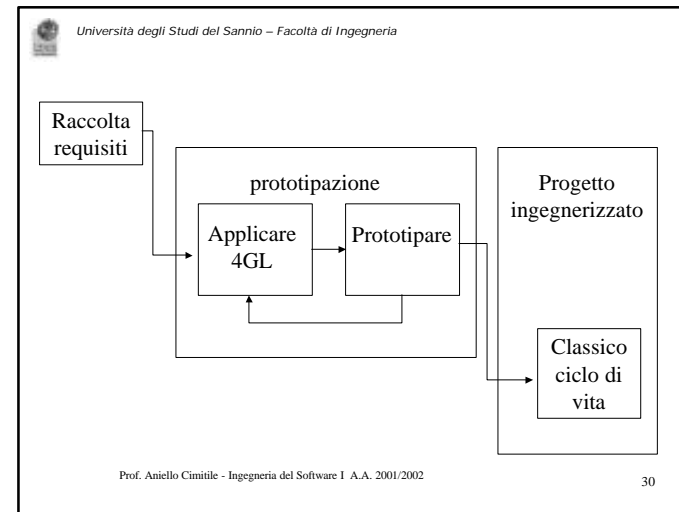
Risorse Tecnologiche

I modelli do it twice richiedono risorse tecnologiche quali:

- Vaste biblioteche di componenti software (ad esempio oggetti)
- Generatori di applicazioni
- Ambienti di quarta generazione
- Generatori di interfacce e linguaggi visuali
- Etc

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

29



Università degli Studi del Sannio – Facoltà di Ingegneria

Rischi

- ❑ Il prototipo è uno strumento di identificazione dei requisiti ... il guaio è che spesso diventa... un prodotto

Lento, troppo grande, di difficile uso, carente funzionalmente, inefficiente, gravi problemi di manutenibilità, etc...

IL PRODOTTO DEVE ESSERE GETTATO

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

31

Università degli Studi del Sannio – Facoltà di Ingegneria

Modelli evolutivi ed incrementali

- ❑ Un principio "SOFTWARE MUST EVOLVE"
- ❑ Superare un limite dei modelli precedenti
Produce "presto" qualcosa di utile, installabile, operativo

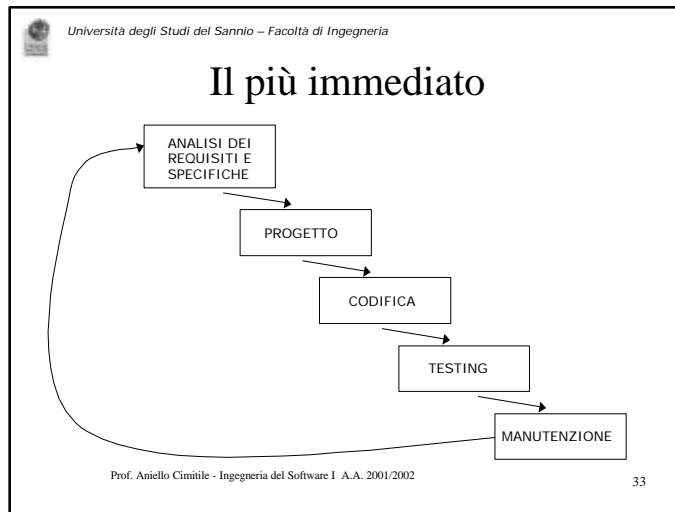
Boehm 1988

EVOLUTIONARY PROCESS MODEL

"...a model whose stages consist of expanding increments of an operational software product, with the direction of evolution being determined by operational experience"

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

32



Università degli Studi del Sannio – Facoltà di Ingegneria

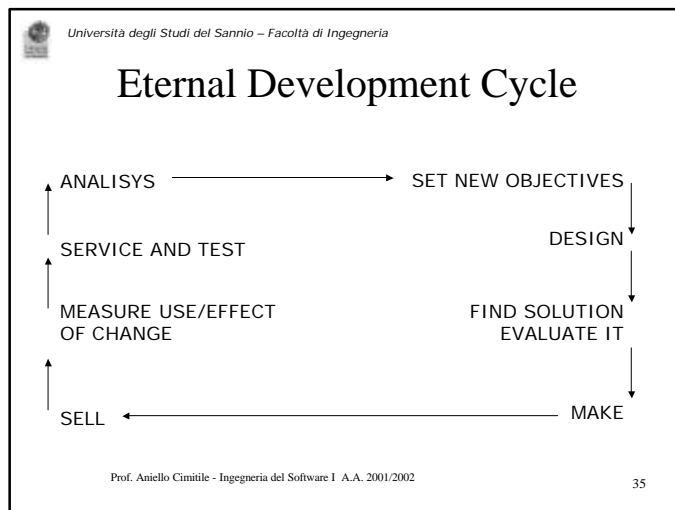
□ Introduzione di

- una apposita fase di EVOLUZIONE
- del ...riciclo
 - Fase di EVOLUZIONE
 - ❖ Analisi dell'esperienza ed utilizzazione della maggiore conoscenza per definire nuovi obiettivi
 - ❖ Nuove esigenze e nuove funzionalità emerse o non coperte in precedenza
 - Riciclo
 - ❖ Sulla base dei risultati della fase di Evoluzione, si riprende il ciclo della definizione dei requisiti alla messa in esercizio e manutenzione del nuovo sistema software nato dall'arricchimento e dalla evoluzione del precedente

E' IN QUESTO MODO CHE SONO STATI PRODOTTI LA MAGGIOR PARTE DEI "LEGACY SYSTEMS" OGGI ESISTENTI

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

34



Università degli Studi del Sannio – Facoltà di Ingegneria

Modello ad Implementazione Incrementale

... dall'accoglimento del principio dell'evoluzione ...

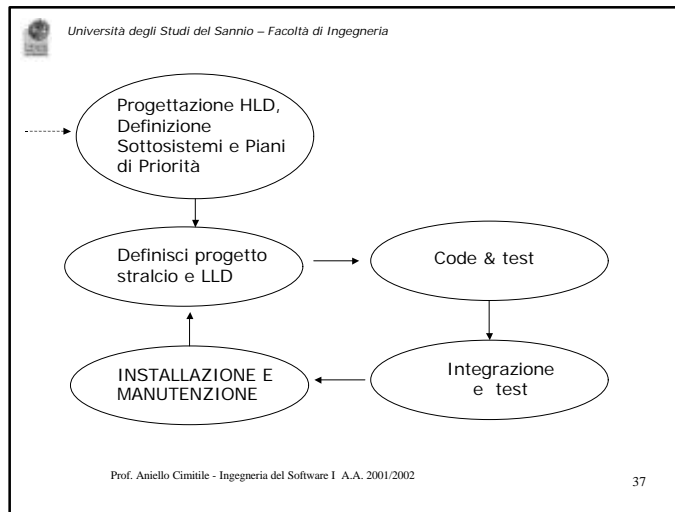
... alla costruzione incrementale di un sistema software con approccio sistematico

- Le fasi alte del WM sono completamente realizzate il sistema software viene completamente definito nei requisiti, specificato e progettato (... solo HLD ...)
- Il sistema progettato viene decomposto in sottosistemi che vengono implementati, testati, deliberati, installati e messi in esercizio e manutenzione secondo un piano di priorità in tempi diversi

Diventa fondamentale una fase (o un insieme di attività) di integrazione dei nuovi sistemi prodotti con quelli già in esercizio

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

36



Università degli Studi del Sannio – Facoltà di Ingegneria

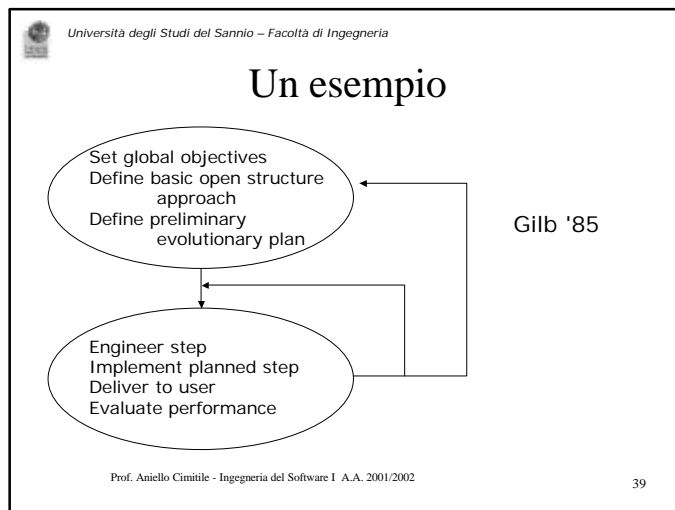
Considerazioni

- ❑ Risponde ad una esigenza del produttore, ... ma può anche rispondere ad una esigenza del committente
- ❑ Molto diffuso in Italia nella PA e negli Enti Locali

... vedi suggerimenti alla formulazione di piani triennali

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

38



Università degli Studi del Sannio – Facoltà di Ingegneria

Un esempio: S&S Model

- Synchronize-and-Stabilize Model
 - Definito per Microsoft, Inc. da Cusumano e Selby (1997)

n.b.Windows 95: 11 Milioni di LOC
200 programmatori
Più teams (3-8 programmatori + 3-8 testers) al lavoro in parallelo

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

40

Università degli Studi del Sannio – Facoltà di Ingegneria

- **Analisi dei Requisiti**
 - La fase di analisi viene condotta attraverso l'intervista di un elevato numero di potenziali utenti che definiscono anche la priorità delle caratteristiche desiderate
- **Specifiche**
 - Viene prodotto il documento delle Specifiche per tutto il sistema
- **BUILDS**
 - Il lavoro da fare viene suddiviso in più “builds” completamente specificati
- **TEAM WORK**
 - Ciascun build viene affidato ad un team work
 - Ciascun membro del team realizza il “design” e l'implementazione di una parte del build

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

41

Università degli Studi del Sannio – Facoltà di Ingegneria

- **SINCRONIZZAZIONE**
 - Quotidianamente i componenti prodotti (anche parzialmente completi) vengono messi insieme e sottoposti a testing e debugging
 - Selby rule: ciascuno faccia quello che vuole durante la giornata, purchè depositi all'ora stabilita il proprio lavoro nel data base per la sincronizzazione quotidiana (.. E tutti a letto alle 9pm)
- **STABILIZZAZIONE**
 - A conclusione di ogni build, fixing delle anomalie e degli errori rilevati e... “the build is frozen”

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

42

Università degli Studi del Sannio – Facoltà di Ingegneria

Modello a sviluppo e consegna incrementale

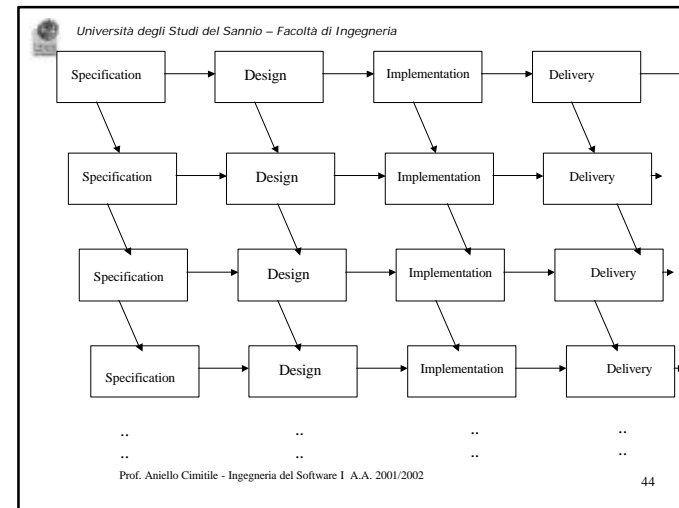
L'approccio incrementale viene esteso a tutte le fasi del ciclo di sviluppo WM

E' uno slicing del processo di produzione dell'intero sistema

- Ogni slice definisce un insieme ben definito di comportamenti del sistema e procede dalla definizione dei requisiti alla implementazione etc... del software che riproduce tali comportamenti
- I prodotti di una nuova slice vengono aggiunti ed integrati con l'esistente e costituiscono un incremento del sistema

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

43



Università degli Studi del Sannio – Facoltà di Ingegneria

□ Anche questo è molto diffuso nella PA di tutto il bisogno software si decide di soddisfare solo una parte oggi, poi, domani un'altra, etc...
es. oggi il bilancio e la contabilità, domani il personale, ...

□ Più "nobilmente" decomposizione delle caratteristiche del sistema in "feature chunks" per la prima slice selezione delle "critical success features" poi ... poi

□ I RISCHI DELL'ANALISI E PROGETTAZIONE SEPARATA
□ IL BISOGNO DI ARCHITETTURE OPEN E DI STANDARD

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

45

Università degli Studi del Sannio – Facoltà di Ingegneria

Modelli a prototipazione evolutiva

□ Produzione evolutiva del prototipo
□ Trasformazione evolutiva del prototipo nel sistema finale

Riportato a scala reale
Completamento funzionalità
Eliminazione stubs e simulazioni
Reingegnerizzazione e raffinamento soluzioni grossolane

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

46

Università degli Studi del Sannio – Facoltà di Ingegneria

CVS per sistemi esperti

... non si conosce a priori a che cosa andiamo incontro
Cosa dobbiamo progettare ed implementare viene scoperto ... per la strada ...

E' vitale il feedback permanente fra tutti i passi del processo di sviluppo qualunque siano

Attori: Esperto del dominio, Ing. della Conoscenza, Ing. del Software

Approccio necessariamente evolutivo

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

47

Università degli Studi del Sannio – Facoltà di Ingegneria

```
graph TD; I[IDENTIFICAZIONE] <--> C[CONCETTUALIZZAZIONE]; I <--> T[TESTING]; I <--> F[FORMALIZZAZIONE]; I <--> IM[IMPLEMENTAZIONE]; C <--> T; C <--> F; C <--> IM; T <--> IM; F <--> IM;
```

CVS per Sistemi Esperti
Rete Completamente Connessa di Fasi

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

48

Università degli Studi del Sannio – Facoltà di Ingegneria

Modello Trasformatzionale

Lo sviluppo del Software viene visto come una sequenza di passi che trasformano formalmente una specifica in una implementazione

- **Primo passo:** analisi dei requisiti informali e loro trasformazione in specifiche formali
- **Ciclo:** dalle specifiche formali a specifiche formali di più basso livello
- **Fino al:** codice eseguibile

Ad ogni livello di specifiche formali corrisponde un processore astratto in grado di eseguirle

Trasformazioni Manuali
Trasformazioni interattive Macchina-Ing. del Software
Trasformazioni Automatiche

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

49

Università degli Studi del Sannio – Facoltà di Ingegneria

I prodotti del CVS sono tre:

- Le Specifiche Formali
- Il Sistema da rilasciare
- Diario dello Sviluppo

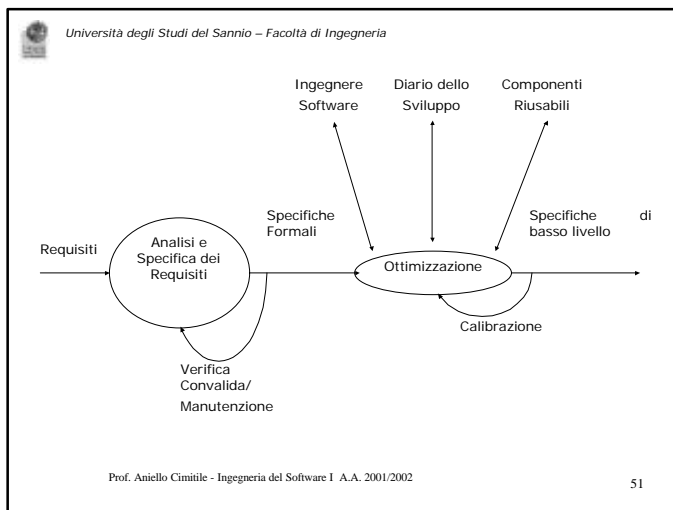
Insieme delle regole di trasformazione applicato ad ogni livello di dettagliamento delle specifiche

Specifiche formali e Regole di trasformazione sono la base per la manutenzione

Rilevanza di supporti automatici e ruolo centrale di una repository di componenti riusabili

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

50



Università degli Studi del Sannio – Facoltà di Ingegneria

Considerazioni

Allo stato il modello trasformatzionale non può essere considerato un paradigma pratico per la produzione del software

E' ancora oggetto di ricerca e vi sono pochi ambienti sperimentali di supporto

Studiato per la produzione di piccoli programmi... si cerca di scalarlo verso la produzione di software reale

Si studiano 'Automated Assistant', sistemi esperti per l'Ing. del Software

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

52

Università degli Studi del Sannio – Facoltà di Ingegneria

The Spiral Model

- È un modello risk-driven
- Applicabile solo a sistemi “large scale” (L, VL, EL)
- Sviluppato esclusivamente per sviluppo interno (sviluppatore e cliente appartengono alla stessa organizzazione), Boehm 1988

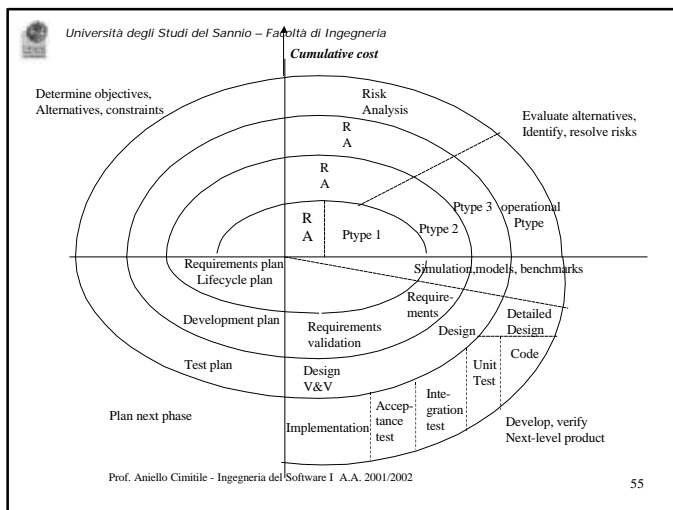
53

Università degli Studi del Sannio – Facoltà di Ingegneria

- Rapid prototyping finalizzato alla minimizzazione dei rischi
- Risk analysis: è il primo task in ogni fase di sviluppo
- Misure, stime, valutazioni...e modelli, metodi, tecniche e tools per misurare, stimare, valutare.....

Fattore critico: l'esistenza nei teams di sviluppo di “risk analysts” competenti ed esperti

54



Università degli Studi del Sannio – Facoltà di Ingegneria

- Raggio Spirale: Costi Accumulati
- Angolo: Stato di avanzamento del processo
- La manutenzione è ..un altro ciclo della spirale
- Boehm 1988: applicato a 25 progetti è stata registrata una decisa crescita di produttività (almeno il 50% e fino al 100%)

56

Università degli Studi del Sannio – Facoltà di Ingegneria

Reuse Model

- Utilizzare componenti software già esistenti per la produzione di un nuovo sistema
- Riutilizzo accidentale
- Riutilizzo Organizzato e Sistemico
– Le “SOFTWARE FACTORY”

57

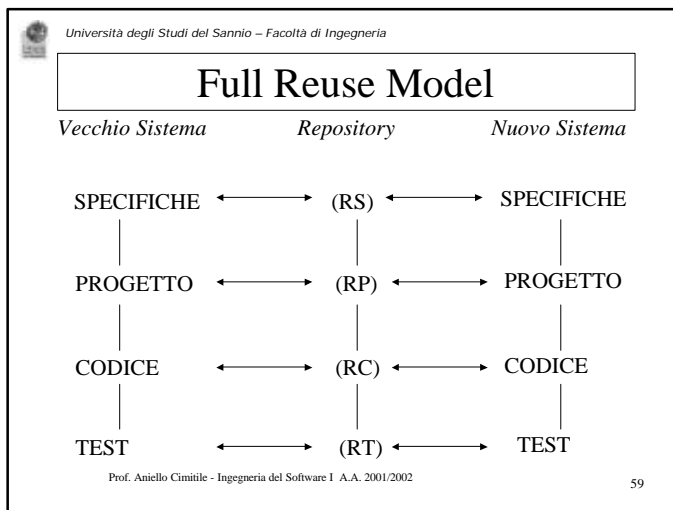
Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

Università degli Studi del Sannio – Facoltà di Ingegneria

- Il riutilizzo nel codice: frammenti di codice, subroutine, le librerie di subroutine, i moduli,.....
- Come per i componenti hardware.....avere “chips” software...
- 1° problema: componenti riutilizzabili
- 2° problema: supporti per la archiviazione e ricerca
- 3° problema: ambienti di supporto alla interoperabilità

58

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002



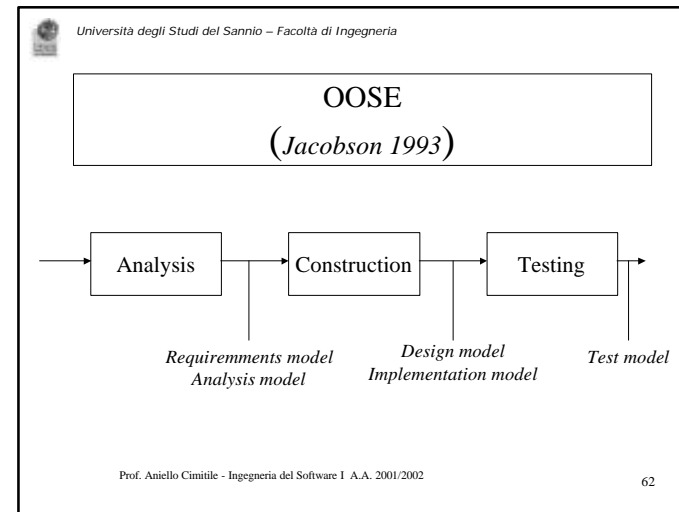
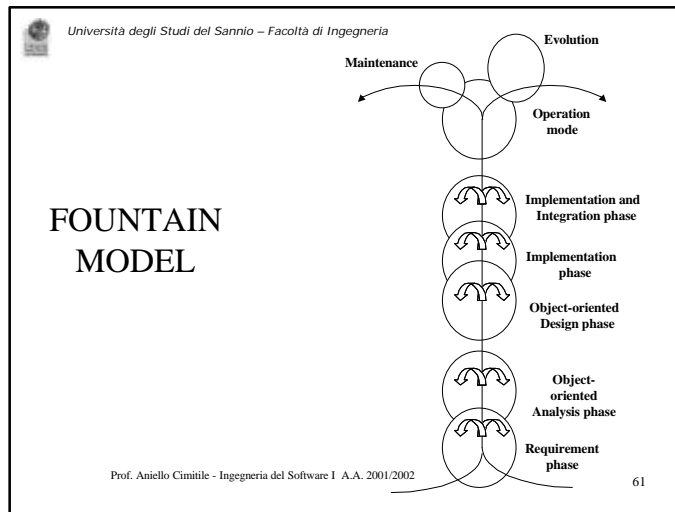
Università degli Studi del Sannio – Facoltà di Ingegneria

Dal Paradigma Strutturato al Paradigma Object Oriented

<ul style="list-style-type: none"> • Requirements phase • Specification phase • Design phase • Implementation phase • Integration phase • Maintenance phase • Retirement 	<ul style="list-style-type: none"> • Requirement phase • O-O analysis phase • O-O design phase • O-O programming • Integration phase • Maintenance phase • Retirement
---	--

60

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002



Università degli Studi del Sannio – Facoltà di Ingegneria

....altri modelli...


- Recursive/Parallel lifecycle (Berard 1993)
- Round-trip gestalt design (Booch 1994)
-

Tutti sono iterativi , prevedono forme di overlap delle attività, supportano lo sviluppo incrementale

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002

63

- Università degli Studi del Sannio – Facoltà di Ingegneria
1. Non trasformare l'esigenza di iterazioni e raffinamenti in un approccio indiscriminato allo sviluppo del software: CABTAB (..Code a bit Test a bit..)
 2. Numerose classi di metodi e modelli (OOD/Booch, OOSA/Shlaer & Mellor, OOA/Coad & Yourdon, HOOD/esa, OOSD/Wassermann, OMT/Rumbaugh,.....) fino alla nascita di un linguaggio unificato di modellazione:UML
- Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002
- 64

 *Università degli Studi del Sannio – Facoltà di Ingegneria*

- La manutenzione: inglobata nel modello, perché ...concettualmente equivalente alla produzione di un nuovo sistema....
- La produzione di componenti riusabili : molto costosa (Tracz 1994, +60%)
- Fattore di successo: le repositories
- Gli aspetti legali...

Prof. Aniello Cimitile - Ingegneria del Software I A.A. 2001/2002 65