

 Considerazioni strategiche

Considerazioni strategiche

IS

Anno accademico 2014/15
Ingegneria del Software mod. A

Tullio Vardanega, tullio.vardanega@math.unipd.it

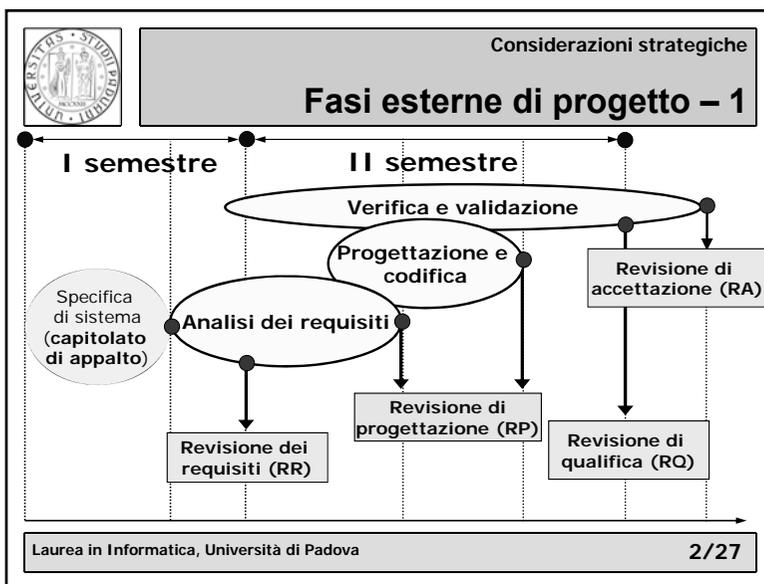
Laurea in Informatica, Università di Padova 1/27

 Considerazioni strategiche

Gradi di libertà

- Il segmento di ciclo di vita attivato nel progetto didattico è compreso tra RR e RA
 - Revisione dei requisiti (inizio)
 - Revisione di accettazione (collaudo, fine)
- Il modello di sviluppo interno a esso
 - È scelta autonoma del fornitore
 - Non è specificato nel capitolato di appalto
 - Determina piano e strategia di utilizzo delle risorse
 - Persone, procedure, strumenti

Laurea in Informatica, Università di Padova 3/27



 Considerazioni strategiche

Pianificazione – 1

- Compiti, risorse e tempo necessario
 - *The Mythical Man-Month*, Frederick P Brooks, Jr (1975)
 - Vi sono componenti di impegno non comprimibili
 - Sulle quali non si può accelerare
 - Vi sono compiti non partizionabili
 - Per necessità: una sola istanza di risorsa critica (p.es. ambiente di prova)
 - Per scelta: per preservare integrità concettuale (in analisi e progettazione)
 - Alcuni compiti richiedono comunicazione e interazione
 - Coordinarsi troppo frequentemente costa molto sforzo
 - Le verifiche a livello sistema si fanno solo alla fine
 - Il sistema (completo) diventa disponibile solo a fine sviluppo

Laurea in Informatica, Università di Padova 4/27

Considerazioni strategiche

Pianificazione – 2

- ❑ **Aggiungere risorse a progetto in corso**
 - Aggiunge complessità nell'esecuzione dei compiti che richiedono comunicazione e coordinamento
 - Aggiunge inefficienza nell'esecuzione dei compiti non partizionabili
- ❑ **Buona pianificazione, buona analisi e buona progettazione**
 - Richiedono investimento a monte, quando c'è tempo
 - Risparmiano costo a valle, quando le risorse mancano

Laurea in Informatica, Università di Padova 5/27

Considerazioni strategiche

Modello sequenziale – 1

RR RP_{min} RP_{max} RQ RA

I Specifica Definizione Qualifica Accettazione F

● Sincronizzazione bloccante
● Limiti del segmento di ciclo di vita coperto dal progetto

□ Revisione obbligatoria □ Revisione a scelta

■ Fase/Attività

Laurea in Informatica, Università di Padova 7/27

Considerazioni strategiche

Modello di ciclo di vita interno – 1

- ❑ **Non tutti i modelli si adattano allo stesso modo agli adempimenti formali richiesti dal progetto**
- ❑ **La scelta del modello di ciclo di vita interno è libera e autonoma da parte del fornitore**
 - Ogni scelta comporta oneri e benefici diversi
- ❑ **La scelta è spesso per prodotto (per progetto)**
 - Indipendente dall'organizzazione di appartenenza

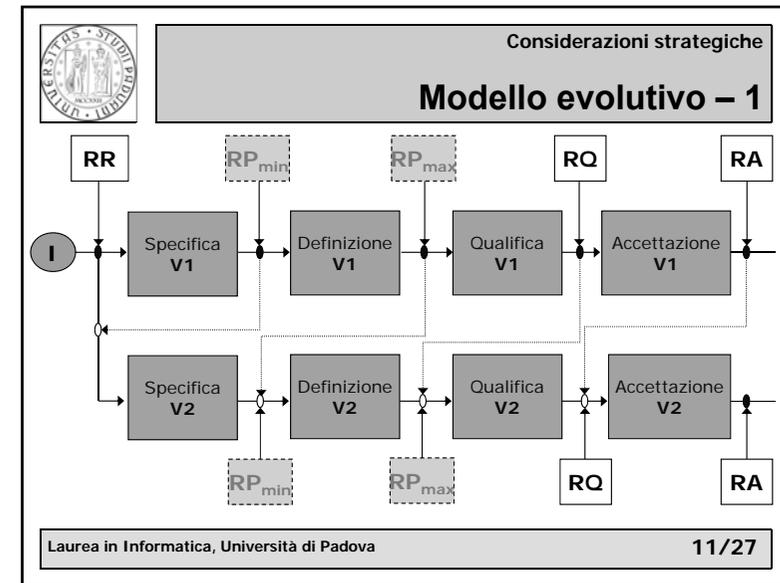
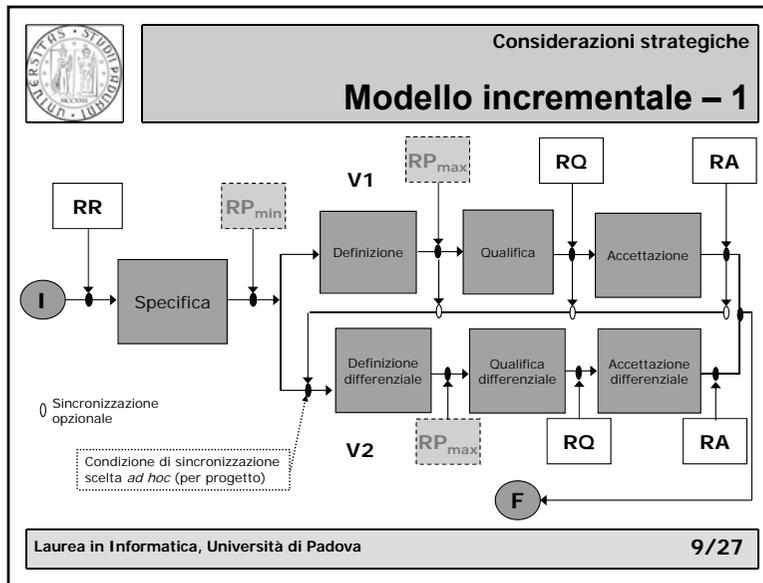
Laurea in Informatica, Università di Padova 6/27

Considerazioni strategiche

Modello sequenziale – 2

- ❑ **Osservazioni**
 - **Il progetto didattico prevede una singola occorrenza di ogni revisione di avanzamento**
 - La sequenza di revisioni rappresenta il punto di vista del committente sull'avanzamento del progetto
 - **Il fornitore può liberamente ritenere le revisioni solo come vincolo di calendario**
 - Senza obbligo di aderire al modello sequenziale per il proprio sviluppo interno
 - A condizione di sviluppare in tempo tutti i prodotti richiesti in ingresso dalle revisioni sostenute

Laurea in Informatica, Università di Padova 8/27



Considerazioni strategiche

Modello incrementale – 2

□ Osservazioni

- Il progetto didattico prevede una sola accettazione
 - Per il committente si tratta del collaudo finale
- Il modello incrementale può però necessitare di più istanze di una stessa revisione di avanzamento per incrementi successivi
- Il fornitore deve decidere quale istanza far corrispondere al calendario di revisione fissato dal cliente
 - Realizzando le altre verifiche di avanzamento come revisioni interne senza il coinvolgimento del committente
 - Le revisioni esterne non possono essere differenziali!

Laurea in Informatica, Università di Padova 10/27

Considerazioni strategiche

Modello evolutivo – 2

□ Osservazioni

- Valgono le stesse considerazioni del modello incrementale
- Ma versioni successive possono rilasciare prototipi esterni e anche essere iterative (= distruttive)
- La combinazione di sviluppi evolutivi con il calendario di revisioni imposti dal cliente ha conseguenze importanti
 - Revisioni successive possono avere come oggetto versioni di prodotto diverse
- I prodotti presentati in ingresso alla revisione devono essere resi coerenti con il fine della revisione (per soddisfare il committente) e l'interesse strategico del fornitore

Laurea in Informatica, Università di Padova 12/27

Considerazioni strategiche

Una visione critica

Rischi

Funzionalità "visibili"

Modello sequenziale

Analisi dei requisiti Progettazione Codifica Qualifica

Rischi

Funzionalità "visibili"

Modello incrementale

Analisi dei requisiti Incremento 1 Incremento 2 Incremento 3

Laurea in Informatica, Università di Padova 13/27

Considerazioni strategiche

Tecniche progettuali – 1

- **Decomposizione modulare**
 - Una buona decomposizione architeturale identifica componenti tra loro indipendenti
 - A basso o nullo accoppiamento
 - Autosufficienti (funzionalmente coesi)
- **Incapsulazione (*information hiding*)**
 - Nascondere il dettaglio realizzativo
 - Solo l'interfaccia è pubblica
 - Il dettaglio è noto solo all'interno

Laurea in Informatica, Università di Padova 15/27

Considerazioni strategiche

Avvertenze

- **Limiti intrinseci della progettazione**
 - Non tutti i problemi hanno una (buona) soluzione
 - Per ogni scelta progettuale serve fissare con la massima chiarezza possibile
 - Obiettivi
 - Vincoli
 - Alternative
 - Rappresentazioni del problema e delle sue soluzioni
- **Qualità cardine della progettazione**
 - Fattibilità e verificabilità
 - Sul piano tecnico e su quello economico

Laurea in Informatica, Università di Padova 14/27

Considerazioni strategiche

Tecniche progettuali – 2

- **Controllo di accoppiamento e di coesione**
- **L'accoppiamento è misura dell'intensità della relazione tra parti distinte**
 - La modifica di una comporta la modifica dell'altra
 - Forte accoppiamento → scarsa modularità
- **La coesione è misura dell'intensità della relazione all'interno di una singola parte**
 - Forte coesione → buona caratterizzazione

Laurea in Informatica, Università di Padova 16/27



Considerazioni strategiche

Tecniche progettuali – 3

- ❑ **Astrazione**
 - **Omettere informazione per poter applicare operazioni simili a entità diverse**
 - La radice di una gerarchia di classi astrae rispetto alle classi più specializzate
 - Ciò che è caratteristico dell'intera gerarchia è fissato in radice
 - Ciò che differenzia si aggiunge per specializzazione
 - **A ogni astrazione corrisponde una concretizzazione**
 - Per parametrizzazione (esempio: da *template* a entità concreta in C++)
 - Per specializzazione (esempio: da interfaccia a classe in Java e C++)
 - Per valorizzazione (esempio: da classe a oggetto tramite costruttore)

Laurea in Informatica, Università di Padova17/27



Considerazioni strategiche

Problematiche critiche – 1

- ❑ **Concorrenza**
 - **Se e come decomporre il sistema in entità attive concorrenti garantendo**
 - **Efficienza** di esecuzione
 - **Atomicità** di azione
 - **Consistenza e integrità** di dati condivisi
 - Semantica precisa di comunicazione e sincronizzazione
 - **Predicibilità** di ordinamento temporale
- ❑ **Distribuzione**
 - **Se e come i componenti sono disseminati su più nodi di elaborazione e come comunicano fra loro**

Laurea in Informatica, Università di Padova19/27



Considerazioni strategiche

Tecniche progettuali – 4

- ❑ **Sufficienza**
 - **La definizione dell'astrazione data è sufficiente a caratterizzare l'entità desiderata**
- ❑ **Completezza**
 - **L'astrazione fissata esibisce tutte le caratteristiche di interesse del suo utilizzatore**
- ❑ **Atomicità**
 - **L'utilità dell'astrazione non migliora se ulteriormente decomposta in astrazioni più elementari**

Laurea in Informatica, Università di Padova18/27



Considerazioni strategiche

Problematiche critiche – 2

- ❑ **Controllo e trattamento degli eventi e degli errori**
 - **Relativi al flusso dei dati**
 - La disponibilità di un dato (dall'interno o dall'esterno)
 - **Relativi al flusso di controllo**
 - L'ingresso del sistema (o di una sua componente) in un particolare stato
 - **Relativi al trascorrere del tempo**
 - L'attesa di un certo istante temporale
- ❑ **Mai fare assunzioni ottimistiche!**

Laurea in Informatica, Università di Padova20/27



Considerazioni strategiche

Integrità concettuale – 1

- ❑ **Facilmente riconoscibile in una architettura fisica (edificio, costruzione, ...)**
 - Suggestisce uno stile uniforme, coerentemente applicato a tutte le parti del sistema ed alle loro interazioni
 - Bilancia capacità funzionale con semplicità d'uso

- ❑ **Desiderabile in ogni architettura di sistema**

Laurea in Informatica, Università di Padova21/27



Considerazioni strategiche

Enforce intentions

- ❑ **Precauzioni da usare al confine tra progettazione e codifica**
 - Rendere chiaro il confine tra esterno e interno dei moduli
 - Decidere chiaramente e codificare coerentemente ciò che può essere specializzato
 - Rendere il resto immodificabile (*final, const, ...*)
 - Proteggere tutto ciò che non deve essere visto e acceduto dall'esterno
 - *Private, protected, ...*
 - Decidere quali classi possono produrre istanze e quali no
 - Usare il *pattern* Singleton per le classi a istanza singola

Laurea in Informatica, Università di Padova23/27



Considerazioni strategiche

Integrità concettuale – 2

- ❑ **Procede da una definizione unitaria**
 - Ma non unilaterale perché passa al vaglio di altri membri del progetto
 - Richiede osservanza ai costruttori e vigilanza all'architetto
 - Nozione aristocratica piuttosto che democratica

- ❑ **È distinta dalla realizzazione concreta**
 - Consente più percorsi realizzativi
 - Facilita parallelismo nella realizzazione ←

Laurea in Informatica, Università di Padova22/27



Considerazioni strategiche

Programmazione difensiva

- ❑ **Programmare esplicitamente il trattamento dei possibili errori**
 - Errori nei dati in ingresso
 - Verificare la legalità dei dati prima di usarli
 - Errori logici
 - Fissare e verificare invarianti, pre- e post-condizioni

- ❑ **La strategia di trattamento (*error handling*) va prevista nella progettazione**

Laurea in Informatica, Università di Padova24/27



Considerazioni strategiche

Errori nei dati in ingresso

- ❑ **Tecniche di trattamento**
 - Attendere fino all'arrivo di un valore legale
 - Assegnare un valore predefinito (*default*)
 - Usare un valore precedente
 - Registrare l'errore in un *log* persistente
 - Sollevare una eccezione
 - Abbandonare il programma

Laurea in Informatica, Università di Padova25/27



Considerazioni strategiche

Fonti di errore logico – 2

- ❑ **Allocazione dinamica della memoria**
 - Può portare a esaurimento della disponibilità e alla sovrapposizione di aree sensibili
- ❑ **Ricorsione**
 - Può portare a esaurimento della memoria oppure alla non terminazione
- ❑ **Concorrenza**
 - Se mal progettata può condurre a situazioni di malfunzionamento difficili da rilevare

Laurea in Informatica, Università di Padova27/27



Considerazioni strategiche

Fonti di errore logico – 1

- ❑ **Aritmetica in virgola mobile**
 - Intrinsecamente imprecisa
 - La sua approssimazione può cumularsi e condurre a errori importanti e a confronti svianti
- ❑ **Puntatori e limiti di strutture**
 - L'accesso erroneo e non controllato a zone di memoria può corrompere gravemente i dati
 - L'uso di *aliasing* rende i programmi difficili da comprendere e modificare

Laurea in Informatica, Università di Padova26/27