

> FSP

> intro

I classici modelli di processo concentrano tutti i rischi nella fase di produzione, per i prodotti hardware questa metodologia è particolarmente indicata perché la fase di riproduzione non presenterà sorprese. Diverso è per prodotti software, dove la fase di riproduzione implica caratteristiche uniche che differenziano un prodotto da un altro.

I classici modelli riscuotono successo nello sviluppo hardware dove non è necessario considerare parametri caratteristici dei sistemi di sviluppo software, non è necessario prevedere la possibilità di rieseguire una fase ad un evento anomalo che causa l'interruzione della linea principale, non è necessario trattare condizioni booleane e non sono necessarie informazioni per descrivere gli eventi che si verificano durante l'esecuzione delle attività.

Anche con le Reti di Petri non abbiamo strumenti adeguati per lo sviluppo di prodotti software, mancano delle caratteristiche importanti come la tracciabilità tra codice e requisiti e l'allocazione delle responsabilità nel caso di mancato soddisfacimento dei requisiti.

Sulla base dei punti di debolezza analizzati sulle Reti di Petri, è stato sviluppato un altro modello, il Design Net Model. Da questo deriva lo Structured Planning. Estensione dello Structured Planning è FSP, in cui si cerca di associare il rigore formale ad una rappresentazione amichevole. Questo nuovo approccio si basa sulla consapevolezza che un progetto deve essere comprensibile prima di tutto alle persone che devono lavorare.

> nota: differenza tra produzione e riproduzione

Per produzione si intende il classico significato, diverso è per riproduzione in cui si aggiunge che il prodotto sviluppato possiede sempre caratteristiche uniche e non è possibile disporre di un sistema deterministico per la pianificazione degli imprevisti nell'implementazione di un modello, questo vuol dire che ogni prodotto software è soggetto a rischi esterni ad esso.

In conclusione, il senso di questa distinzione è che il modello di processo da adottare deve essere adattabile ai rischi sia in fase di produzione che in fase di riproduzione.

> collegamento

FSP è un linguaggio di modellazione nato nel SerLab che gode di tracciabilità e leggibilità, descrive artefatti e attività attraverso diagrammi di flusso e scenari procedurali.

Ogni scenario procedurale descrive una macro attività, o attività elementare, che corrisponde ad una serie di attività base dove ogni attività base è descritta attraverso un certo diagramma di flusso formale. Ogni attività base usa e produce manufatti, per ogni manufatto abbiamo variabili di processo mentre per ogni attività abbiamo verifica e validazione.

Il modello deriva da Structured Planning, estende la sua rigidità a potenzialmente tutte le rappresentazioni strutturate esistenti estromettendo una specifica rappresentazione formale.

La progettazione si sviluppa in tre fasi: la definizione di un modello di processo, la descrizione di un piano di progetto e l'attuazione di un piano esecutivo.

La descrizione delle tre fasi è strutturata per agevolare la comprensione di chi lavorerà sia in lettura che in modifica.

> nota bene: differenza tra attività base ed attività elementare

L'attività elementare è composta da un insieme di attività base, l'attività base viene direttamente dal workflow e mediante un insieme di attività base lo scenario dettaglia un'attività elementare.

> sintesi

Vediamo quindi il modello di processo e le sue componenti nel dettaglio, come manufatti, workflow diagram, scenari procedurali ed attributi.

Vediamo le verifiche sul modello, sia statiche che dinamiche.

Vediamo infine le trasformazioni da modello di processo in piano di progetto, e da piano di progetto in piano esecutivo.

> definizione del modello di processo

> intro

In un modello di processo abbiamo la definizione di manufatti, workflow diagram, scenari procedurali ed attributi.

Si parte da manufatti ed azioni, ad un livello immediatamente superiore abbiamo i workflow diagram che mettono in relazione manufatti ed azioni per costituire l'attività base, mentre ad un livello ancora superiore abbiamo gli scenari procedurali che mettono in relazione le varie attività base con i manufatti scambiati tra queste per costituire l'attività elementare.

Il piano di progetto si basa sull'output del modello di processo, eredita quindi descrizioni di manufatti, variabili di processo e scenari.

Il piano esecutivo restituisce una descrizione del piano esecutivo, lavora sulle attività elementari e restituisce un progetto che gode di piena tracciabilità risolvendo eventuali problemi di indeterminazione presenti nel piano di progetto.

Le descrizioni di modello di processo, piano di progetto e piano esecutivo sono strutturate, il che semplifica operazioni di lettura, comprensione e modifica.

> i manufatti

I manufatti sono essenzialmente documenti, descrivono prodotti o documenti finiti o semifiniti che sono input o output per le attività del sistema o che in generale che hanno pertinenza con il sistema da sviluppare. Il sistema è considerato l'insieme di tutti i manufatti.

Ha senso considerare anche operazioni, come concatenazione alternanza e ripetizione.

La concatenazione $a+b+c+d$, indica che dati due o più documenti, essi sono tutti presenti nella descrizione di un manufatto ed in particolare che ognuno segue il precedente, vale la proprietà associativa.

L'alternanza $a/b/c/d$, indica che dati due o più documenti, solo uno di essi sarà presente nella descrizione del manufatto, valgono le proprietà associativa e commutativa, ed anche la distributiva rispetto alla concatenazione.

La ripetizione $m\{ \}n$, invece indica che un documento può comparire almeno m volte al più n volte nella descrizione del manufatto in cui compare.

> albero di decomposizione dei manufatti

Considerate le operazioni, ha senso parlare di albero di decomposizione di un manufatto come un formalismo gerarchico per la rappresentazione di espressioni costruite con manufatti ed operatori. Abbiamo una struttura ad albero, dove il manufatto prodotto od utilizzato è collocato alla radice dell'albero, mentre sui nodi intermedi e finali sono collocati i manufatti etichettati nell'espressione a rappresentare.

> workflow diagram

I workflow diagram sono diagrammi di flusso formali per la rappresentazione dei dati, si basano sui formalismi dei digrammi di flusso classici uniti ad un carattere comportamentale delle reti di

Petri. I workflow sono 4ple $W = (D_W, T, I, O)$ dove:

- D_W è un sottoinsieme dei manufatti etichettati in W ;
- T è un insieme di attività da esaminare;
- I ed O sono due funzioni che definiscono rispettivamente input ed output del workflow, in base ad un'attività restituiscono $I:T \rightarrow E; O:T \rightarrow E$, dove E è l'insieme delle espressioni booleane basate su AND OR e costruite sull'insieme dei manufatti D .

> albero di decomposizione di un workflow

Descrive un problema complesso mediante decomposizione in workflow in cui ognuno dettaglia un'attività in un certo livello di astrazione. Un'attività T_i sarà descritta dal workflow $W_i = (D_i, T_j, I, O)$ dove le attività T_j sono le attività del workflow ad un livello inferiore di astrazione descritte dal workflow W_j .

La rappresentazione è con la struttura ad albero chiamata Work Flow System WFS, in cui alla radice c'è l'intero progetto mentre sui nodi intermedi e finali ci sono le attività descritte connesse mediante archi o relazioni di padre-figlio.

> scenari procedurali

Il WFS è il concetto che lega workflow e scenari procedurali, per ogni WFS possiamo avere uno scenario procedurale che descrive l'attività elementare attraverso un insieme di attività base.

Gli scenari procedurali sono 5ple $PD = (isc, iec, D_p, B_p, OP_p)$ dove:

- isc, iec sono delle condizioni interne di inizio e fine, vengono introdotte per favorire il riutilizzo del codice e facilitarne la lettura. Con queste condizioni interne lo stesso scenario può essere riutilizzato in altri contesti, o con altri processi o con differenti attributi temporali, senza la necessità di cambiargli l'intera struttura ma semplicemente modificando queste condizioni interne. In particolare le condizioni interne di ingresso hanno un carattere booleano e sono in grado di attivare o scenario;
- D_p è l'insieme dei manufatti etichettati nell'attività elementare T , sono compresi i sottoinsiemi $I(T)$ ed $O(T)$, con relativi discendenti nell'albero di decomposizione dei manufatti;
- B_p è l'insieme delle attività base che compongono l'attività elementare dettagliata nello scenario;
- OP_p è l'insieme degli operatori da utilizzare sulle attività base. È possibile definire

operatori di sequenza, alternanza a una o più vie, iterazione condizionata o incondizionata.

> attributi

Gli attributi sono le proprietà caratteristiche di una precisa componente, possono essere valutati sia in fase di definizione del modello che dinamicamente in fase di esecuzione del modello. Si definiscono attributi per ogni manufatto, attività elementare ed attività base:

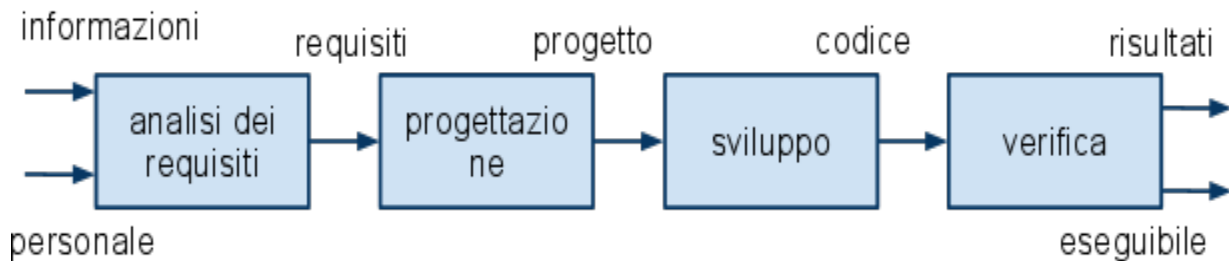
- per manufatti abbiamo metriche, standard, formato, etc;
- per attività elementari o semplicemente attività abbiamo priorità, data inizio, data fine, stato, etc;
- per attività base abbiamo competenze, tools, tecniche, metodi, costo, etc.

Il loro utilizzo risulta vantaggioso per il riuso o la modifica del modello.

> esempio (sarà giusto?)

esempio di progettazione con semplice modello a cascata

- workflow diagram



$$W = (D_w, T, I, O);$$

$$D = (informazioni, personale, requisiti, progetto, codice, risultati, eseguibile);$$

$$T = (analisi dei requisiti, progettazione, sviluppo, verifica);$$

$$I = (analisi dei requisiti; informazioni AND personale),$$

$$(progettazione; requisiti), (sviluppo; progetto), \dots$$

$$O = (analisi dei requisiti; requisiti), (progettazione; progetto), \dots$$

- manufatto

manufatto informazioni:

- a) dati storici organizzazione;
- b) necessità organizzazione;
- c) lista componenti necessarie;
- d) business organizzazione.

- scenario procedurale

scenario procedurale analisi dei requisiti

$$PD = (isc, iec, D_p, B_p, OP_p)$$

*if informazioni AND personale disponibili
 then
 produrre requisiti usando informazioni AND personale
 isc = (informazioni, personale); iec = (nessuna);
 D_p = (informazioni, personale, requisiti);
 B_p = (lettura dati, analisi, test, verifica, validazione); OP_p = nessuno;*

- attributi

attributi manufatto codice: istruzioni di compilazione, standard, ...

attributi attività progettazione: data inizio, data fine, personale, ...

> verifiche

Definiti i componenti si procede con le verifiche sul modello di processo costruito per garantire che non ci siano inconsistenze già a livello di progettazione del processo.

L'operazione di verifica si ricordi rientra nella classe dei processi di supporto, è a cura del produttore e serve principalmente a prevenire aumenti di lavoro nelle fasi più avanzate del ciclo di vita del prodotto software.

Le verifiche possono essere statiche o dinamiche a seconda che vengano eseguite direttamente sulla descrizione di un modello di processo oppure su un modello di simulazione.

> verifiche statiche

Le verifiche statiche operano direttamente sulla descrizione del modello di processo, abbiamo verifiche statiche per manufatti, workflow diagram e scenari procedurali.

- Per manufatti ha senso considerare la verifica di consistenza globale, cioè controllare che uno stesso componente non sia presente più volte nella definizione della struttura. L'algoritmo verifica l'assenza di cicli mappando l'albero di decomposizione con un grafo diretto.
- Per workflow diagram hanno senso le verifiche di consistenza globale e strutturale. La verifica di consistenza globale controlla che una stessa fase non venga richiamata in più livelli in un workflow, mentre la verifica di consistenza strutturale controlla che ogni workflow utilizzi tutti e soli manufatti ed output della fase che dettaglia.
- Per scenari procedurali, le verifiche riguardano sempre le descrizioni dei manufatti ed i workflow, ma potranno essere considerate solo una volta definito lo scenario relativo ad ogni attività elementare menzionata nel processo. Quindi a partire dagli scenari procedurali definiti, si considerano le attività base dei workflow con i relativi manufatti di input ed output per effettuare le verifiche di consistenza. Per scenari procedurali quindi, hanno senso la verifica di consistenza dei manufatti e la verifica di consistenza isomorfica. La verifica di consistenza dei manufatti opera sui workflow utilizzati nello scenario procedurale, in particolare un workflow è consistente rispetto ai manufatti se ogni manufatto etichettato è usato come input da almeno un'attività base del workflow, ed ogni attività base ha sia un manufatto di input che uno di output. La consistenza isomorfica invece lavora tra attività elementare T_0 e scenario procedurale PD_0 che la descrive, e controlla che ci sia isomorfismo tra la struttura dei manufatti di input e

la struttura dello scenario, e tra la struttura dei manufatti di output e la struttura dello scenario. C'è un algoritmo per questo controllo.

Riassumendo sulle verifiche statiche:

- manufatti → consistenza globale;
- workflow diagram → consistenza globale e consistenza strutturale;
- scenari procedurali → consistenza dei manufatti e consistenza isomorfica.

> nota: cos'è l'isomorfismo

L'isomorfismo si verifica quando date due strutture, per ognuna ce n'è sempre una corrispondente nell'altra. Le due parti giocano ruoli simili nelle rispettive strutture.

Dati due grafi, un isomorfismo tra loro è un'applicazione bigettiva cioè che c'è un arco da un vertice all'altro in uno se e solo se c'è un arco analogo nell'altro.

> (pseudo)algoritmo per la consistenza isomorfica

Consideriamo un'attività elementare T_0 e lo scenario che la descrive PD_0 , consideriamo un'unica struttura dei manufatti sia per gli input che per gli output dell'attività quindi sia per $I(T_0)$ che per $O(T_0)$, e per omogeneità nell'algoritmo usiamo il simbolo "+" per l'AND ed il "/" per l'OR rispettivamente per concatenazione ed alternanza.

Ora l'algoritmo mira a controllare tutti i seguenti casi:

1. dato $d = d_1 + d_2 + d_3 + \dots + d_n$, esistono n azioni base che usano o producono tutti gli n documenti etichettati a seconda che d sia input o output;
2. dato $d = d_1 / d_2$, esistono due azioni base in alternativa che usano o producono d_1 e d_2 , *if then else*(b_1, b_2);
3. dato $d = d_1 / d_2 / d_3 / \dots / d_n, n > 2$, esistono n azioni base in alternativa che usano o producono l'i-esimo manufatto presente in d , *select*(b_1, b_2, \dots, b_n);
4. dato $d = 0\{d_1\}1$, c'è un'attività base con esecuzione opzionale, che usa o produce d_1 , *if then*(b_1);
5. dato $d = 0\{d_1\}n, n > 1$, c'è un'attività base con esecuzione opzionale, che usa o produce n volte d_1 , *while*(b_1);
6. dato $d = 1\{d_1\}n, n > 1$, c'è un'attività base che usa o produce un numero condizionato di volte d_1 , *repeat* b_1 ;
7. dato $d = 1\{d_1\}x$, c'è un'attività base che usa o produce un numero incondizionato di volte d_1 , *foreach* (b_1).

> verifiche dinamiche

La verifica dinamica agisce su un modello di animazione del modello descritto, l'aspetto dinamico viene realizzato con le marcature e attivazioni tipici delle reti di Petri.

La marcatura è riferita ai manufatti che saranno input per le attività del processo, si parte con l'analisi del WFS, per ogni attività che si sussegue gli si verranno marcati i manufatti di input

che sono stati già generati per contrassegnarli come abilitati. Un'attività potrà cominciare l'esecuzione quando tutti i manufatti di input saranno abilitati. La fase successiva sarà l'eliminazione delle marcature dai manufatti marcati ed utilizzati dall'attività, e la marcatura dei manufatti prodotti dall'attività.

In questo modo si può controllare l'evoluzione del processo e verificare che nessuna fase utilizzi manufatti non ancora prodotti e non marcati, in casi di anomalie questa tecnica può strozzare l'evoluzione fino ad interromperla fin quando lo sviluppatore non apporta le giuste correzioni.

Sotto questi aspetti la funzione $I(T)$ di una quadrupla di un workflow diventa una funzione booleana che attiva un'azione del WFS.

Hanno senso i controlli di consistenza globale e comportamentale.

> modifiche sul modello di processo

Possiamo avere modifiche al workflow diagram, alla descrizione dei manufatti, agli scenari procedurali o agli attributi che specificano le variabili di processo.

Bisogna considerare come le modifiche impattano sul resto del modello di processo, se si stanno modificando alcune descrizioni dei manufatti sarà importante anche aggiornare tutti gli eventuali scenari che fanno uso di quel manufatto. In genere le modifiche devono essere fatte per massimizzare il riuso dei manufatti.

La tecnica migliore per massimizzare il riuso dei modelli di processo, è mediante appunto, una libreria interna all'organizzazione che contenga tutti i modelli di processo costruiti, con lo scopo di riutilizzare un predefinito workflow per una specifica attività, e dettagliare altre attività elementari mediante gli stessi workflow predefiniti.

> trasformazioni

Finché è possibile, è sempre opportuno sfruttare componenti già pronte dell'organizzazione. E' importante considerare però che non sempre vale la pena riutilizzare sempre un modello già utilizzato, in certe condizioni è più costoso considerare l'enorme numero di attributi piuttosto che definirne uno nuovo. Fermo restando ciò, qualunque sia la scelta è importante che il modello sia verificato prima di poter parlare di trasformazione in piano di progetto e poi piano esecutivo.

> differenza processo progetto piano

I tre stadi sono relativi alla progettazione di un prodotto.

Nel modello di processo viene sviluppata la descrizione concettuale del processo, con la trasformazione in piano di progetto questa descrizione concettuale viene portata in sequenza temporale di attività, si considerano le risorse tutte potenzialmente illimitate e con attività con massimo parallelismo consentito. In fine, nel piano esecutivo si fanno i conti con la realtà dell'organizzazione, viene considerato l'aspetto reale della limitazione delle risorse umane e strumentali, e dell'urgenza delle consegne applicando meccanismi di priorità.

> piano di progetto

> trasformazione da modello di processo in piano di progetto

E' una sequenza temporale di attività relativa ad un frammento del modello di processo, PMF.

Un PMF potrebbe essere anche tutto il modello di processo, in genere si prende come PMF una parte che sia eseguibile in un tempo in cui non si alterino le variabili di processo, come una giornata solare.

Si fa uso del concetto di PMF come uno spezzone di un workflow o scenario procedurale del modello di processo. Un PMF nel suo livello di astrazione vede le attività A_i che concretamente possono essere sia attività base che elementari a seconda che si stia considerando workflow o scenario procedurale.

Formalismi per rappresentare il piano di progetto sono i digrammi PERT e GANTT.

> nota: diagrammi PERT

E' acronimo di Program Evaluation and Review Technique, è un formalismo grafico sviluppato per la progettazione di sottomarini militari con lo scopo di mantenere coordinati migliaia di fornitori e subappaltatori. Ha una rappresentazione reticolare, considera le attività interdipendenti e le risorse a disponibilità infinita.

Sui nodi colloca le attività, nodi N_i attività A_i , ogni nodo ha almeno una freccia entrante e solo una in uscita, ci sono due nodi speciali uno di inizio ed uno di fine.

La funzione logica PERT(x) restituisce il diagramma PERT di un certo input x, ha senso considerare PERT(PMF).

> PERT vs GANTT

Mentre con i diagrammi PERT abbiamo una visione ad albero simile ad un diagramma di flusso, con i nodi connessi mediante frecce, i diagrammi GANTT offrono una rappresentazione ad assi cartesiani con tempo sulle ascisse e attività sulle ordinate.

> nota: la tracciabilità

E' la possibilità di ricostruire la relazione tra diversi documenti prodotti nel corso di un progetto.

E' la possibilità di ricostruire le relazioni degli elementi di un progetto con le specifiche dei requisiti iniziali.

> definizioni importanti

1. Diretta tracciabilità: relazione tra due manufatti in cui uno viene usato e l'altro viene prodotto da una stessa attività.
 - o Una coppia ordinata di manufatti, si dice tracciabile direttamente se esiste un'attività che ne prende in input uno ed in output l'altro.
 - o Dati $(D_i, D_j): \exists A_i \text{ tale che } D_i \in \{I_i\}, D_j \in \{O_i\}$
2. Indiretta Tracciabilità: relazione tra due manufatti in cui uno viene usato e l'altro viene prodotto da una sequenza di attività.
 - o Una coppia ordinata di manufatti, si dice tracciabile indirettamente se esiste una sequenza ordinata di manufatti tracciabili direttamente.
 - o Dati $(D_i, D_j): \exists (D_{i1}, D_{i2}, \dots, D_{in}, D_j) \text{ in cui } \forall k \in [i1, \dots, in] \text{ tale che } (D_k, D_{k+1}) \text{ tracciabile direttamente}$
3. Scope: proprietà che caratterizza un manufatto e rappresenta il numero di manufatti

tracciabili direttamente o indirettamente da questo.

- Ogni manufatto ha una sua area di visibilità detta scope SC_i . Dato un manufatto D_i , il suo SC_i è il numero di manufatti che si possono tracciare direttamente o indirettamente da esso.
- 4. Follows: proprietà che caratterizza un'attività e rappresenta l'esistenza di un legame di output/input tra manufatti rispetto ad un'altra attività sia direttamente che indirettamente.
 - Un'attività segue un'altra se esiste un manufatto che appartiene agli output della prima e contemporaneamente agli input della seconda, questa proprietà si chiama FOLLOWS.
 - Date A_k, A_i ; A_k FOLLOWS A_i se:
 - $\exists D_i$ tale che $D_i \in \{I_i\}$ e $D_i \in \{O_k\}$
 - OPPURE
 - $\exists (D_j, D_p)$ tale che $D_j \in \{I_k\}$ e $D_p \in \{O_i\}$ e (D_j, D_p) tracciabili indirettamente.

> trasformazione di un PMF in PERT

Fondamentalmente l'idea è ad un certo tempo T, sono attivabili tutte quelle attività i quali input sono validi. L'algoritmo massimizza la produttività rendendo parallele tutte quelle attività che possono essere tali.

Ora, partendo da un PMF, vediamo come ottenere il suo PERT(PMF).

1° passo

- inizializza il PMF a START;
- inizializza l'insieme $\{AD\}$ manufatti disponibili con gli input esterni che si suppone essere validi;

2° passo

- per ogni attività A_i del PMF, se i suoi input sono validi cioè appartengono ad $\{AD\}$ e quindi $I(A_i)$ vera, inserisci il nodo N_i ;
- collega N_i con tutti i nodi già tracciati che hanno manufatti di output che sono input per l'attività A_i ;
- aggiorna $\{AD\}$ aggiungendo i manufatti prodotti da A_i ;

3° passo

- ripetere il passo 2 finché per ogni attività A_i del PMF non corrisponda un nodo N_i ;

4° passo

- terminare con il nodo STOP connettendolo a tutti i nodi N_i che non hanno frecce uscenti.

> piano esecutivo

> trasformazione da piano di progetto in piano esecutivo

In questa fase avverranno la maggior parte dei cambiamenti perché bisogna tener conto di tutti

gli eventi ai quali il progetto deve dare conto e che si verificano durante la sua esecuzione. Si utilizza la tecnica a raffinamenti successivi, un raffinamento rappresenta un'evoluzione del piano esecutivo in seguito ad una causa esterna. Inizialmente il piano di progetto ed il piano esecutivo coincidono.

Il piano esecutivo è una rivisitazione del piano di progetto in cui si tiene conto delle cause esterne e si adatta il progetto a queste.

Si fanno uso degli stessi formalismi visti per il piano di progetto, cioè i PERT ed i GANTT.

Nel piano esecutivo il manager viene messo di fronte alla disponibilità delle risorse, se le risorse necessarie alle attività sono tutte disponibili allora il manager potrà eseguirle tutte in parallelo ed il piano esecutivo coinciderà con il piano di progetto. Diversamente dovrà schedulare con una sequenza logica che rispetti la disponibilità delle risorse ed in ordine di priorità. In caso di priorità uguale il manager terrà conto dell'importanza dei manufatti output delle attività sia per importanza per il committente che per lo scope. Il manager potrà adottare anche una di queste due strategie:

- seguire l'efficacia del progetto, tenendo conto delle attività che producono manufatti intermedi più immediatamente tracciabili con i manufatti di output;
- seguire la flessibilità della schedulazione, tenendo conto dello scope dei manufatti.

Conseguenza logica di questi presupposti, è quando le attività sono già in serie nel piano di progetto lo saranno per forza anche nel piano esecutivo per motivi non di risorse ma di progettazione evidentemente, diverso è quando sono in parallelo.

Nei casi in cui non è possibile applicare parallelismo tuttavia si potranno seguire due strade, o decidere di aumentare le risorse ripercorrendo però tutto l'iter di progettazione a partire dal modello di processo, oppure riprendere comunque il modello di processo ed analizzare gli scenari procedurali con una granularità più fine (con informazioni più dettagliate).

> modifiche al piano esecutivo

Può essere necessario applicare modifiche al piano esecutivo in conseguenza ad eventi generati o internamente o esternamente. Gli eventi interni sono causati dal cambiamento spontaneo delle variabili di processo, mentre quelli esterni sono causati dalle iniziative di miglioramento applicate.

Le modifiche comunque avvengano, attraversano una serie di raffinamenti successivi che vengono applicati in seguito a misurazioni che aiutano gli ingegneri a comprendere meglio alcune variabili del processo.

> dinamica del progetto

In parallelo all'evoluzione della progettazione introduciamo gli stati, si potrebbe dire che la decisione di assegnare un nome a qualsiasi risorsa, anche temporanea, coinvolta nel progetto ha un fine principalmente di debugging.

Rappresentiamo quindi la dinamica del progetto con un formalismo che assegna uno stato per ogni oggetto o attività coinvolta. Anche il progetto intero è una risorsa e anche il progetto avrà uno stato.

Il modello vede componenti semplici e componenti composti, quindi assegna stati per componenti semplici e stati per componenti composti. I componenti semplici sono manufatti e attività e per essi avremo stati semplici. Il modello si prepara a considerare casi in cui

l'evoluzione implica contemporaneamente eventi per manufatti ed attività, quindi questi saranno gli stati composti che saranno espressi in funzione di quelli semplici. Dal punto di vista matematico, il modello predispone ogni componente con valori da 1 ad n , assegnando una porzione da 1 a k con $k < n$, per componenti semplici, e da $k+1$ ad n per quelli composti. Quindi avremo stati per attività semplici, stati per manufatti semplici, stati per attività e manufatti composti.

La logica fondamentale è se ho un gruppo di oggetti, affinché io possa assegnare uno stato "disponibile" devo avere per forza un certo sottoinsieme di questo gruppo che è disponibile, affinché io possa assegnare uno stato "in evoluzione" dovrò avere un certo sottoinsieme che è in stato esecuzione. Viceversa, affinché io possa assegnare uno stato "completato" dovrò necessariamente avere tutti gli oggetti nell'insieme considerato, in stato completato. Stesso discorso per stato "scartato".

Le transizioni di stato generano eventi, ogni evento porta il processo in un nuovo stato. Avremo eventi regolari, eventi straordinari ed eventi critici. Per ognuno abbiamo una serie transizioni che muovono il progetto verso la corretta evoluzione.

Nel caso di eventi critici, avremo il modello che si prepara a questa eventualità e predispone anche uno scenario procedurale che lo descrive, in tal caso si parla di attività elementare fuzzy (o sfocata). In questo modo il progetto si prepara ad accogliere l'anomalia e cerca di superarla pianificandone l'esecuzione con un'attività dedicata.

Alcune conseguenze dovute ad eventi critici sono la revisione del piano esecutivo, la ripetizione di alcune attività o l'eliminazione di attività dal piano esecutivo.