

Esame Software Engineering (AA 2021/22)

21 Luglio 2022

Enrico Tronci

Computer Science Department, Sapienza University of Rome
Via Salaria 113 - 00198 Roma - Italy

tronci@di.uniroma1.it

<http://mclab.di.uniroma1.it>

Esercizio 1 (25 punti)

L'unità di tempo per il presente esercizio è il giorno.

Un'azienda ha un ciclo di sviluppo software consistente in N fasi (numerate da 1 ad N e dove N è la fase di delivery dopo la quale si inizia un nuovo progetto).

Il personale dell'azienda consiste di W teams di sviluppo.

I valori N e W sono parametri del modello.

Dai dati storici si vede che il tempo atteso (in giorni) $\tau(i, k)$ per il completamento delle fasi $i = 1, \dots, N$ da parte del team $k = 1, \dots, W$ è:

1. $\tau(i, k) = \text{if } A(i, k) \text{ then } 10 \text{ else } 50.$
2. $A(i, k) = \text{if } (i \% W + 1 = k) \text{ then } 1 \text{ else } 0.$

Modellare il ciclo di sviluppo con una *Discrete Time Markov Chain* (DTMC) con N stati corrispondenti alle diverse fasi del ciclo di sviluppo. L'elemento $p_{i,j}(k)$ della matrice di transizione $P(k)$ della DTMC per il team k è definito come segue.

Quando una fase i è completata, si passa alla fase successiva $i + 1$ con probabilità $p_{i,i+1}(k)$ ovvero, a causa di rilevazione di errori nel progetto, si torna in una delle fasi precedenti $j < i$ con probabilità $p_{i,j}(k)$.

Fa eccezione la fase di delivery (N), dalla quale si transisce sempre con probabilità 1 alla fase 1, cioè si inizia un nuovo progetto con le stesse caratteristiche del precedente. Quindi $p_{N,1}(k) = 1$ per ogni k , cioè per ogni team.

Si ricordi che se $p_{i,i}(k)$ è la probabilità di rimanere nello stato i della DTMC allora il numero atteso $\theta(i, k)$ di transizioni prima di lasciare i è:

$$\theta(i, k) = \frac{1}{1 - p_{i,i}(k)}$$

Quindi, se T è il time step della DTMC, allora il tempo atteso di soggiorno nello stato i (cioè il tempo atteso di completamento della fase i) è $\tau(i, k) = T\theta(i, k)$.

Potete assumere $T = 1$. Cioè il time step è un giorno.

Su questa base e dai dati storici per i tempi di completamento delle varie fasi è possibile calcolare le probabilità $p_{i,i}(k)$.

Per le altre probabilità, dai dati storici si hanno le seguenti relazioni per il team k :

1. $p_{1,2}(k) = 1 - p_{1,1}(k)$
2. Per $i = 2, \dots, N - 1$ si ha: $p_{i,i+1}(k) = (1 - \alpha(k))(1 - p_{i,i}(k))$
3. Per $i = 2, \dots, N - 1$, per $j = 1, \dots, i - 1$ si ha: $p_{i,j}(k) = \alpha(k) \frac{1 - p_{i,i}(k)}{(i-1)}$.
4. $\alpha(k) = \frac{1}{5+k}$
5. Tutte le altre probabilità hanno valore 0.

Il costo giornaliero (in Eur) $C(k)$ del team $k = 1, \dots, W$ è:

- $C(k) = 500 + 1000 \frac{k-1}{W-1}$

Il tempo necessario per completare un progetto (tempo di completamento) è il tempo necessario per arrivare alla fase N (delivery) partendo dalla fase 1 (requirements analysis).

Il costo per completare un progetto è dato dal tempo di completamento moltiplicato per il costo $C(k)$ del team usato nello sviluppo.

Se sviluppi un modello Modelica che per ogni team calcoli il tempo ed il costo atteso di completamento di un progetto. Il costo atteso è il tempo atteso (in giorni) moltiplicato il costo giornaliero del team.

Il vostro modello conterrà:

1. Blocco **DTMC** nel file `dtmc.mo` che modella la DTMC che corrisponde al ciclo di sviluppo descritto sopra.
2. Blocco **Monitor** nel file `monitor.mo` che calcola:
 - (a) il valore atteso del tempo e del costo di completamento di un progetto per ogni team
 - (b) la deviazione standard del tempo e del costo di completamento di un progetto per ogni team

Si usi l'istruzione Modelica **terminate** per terminare la simulazione quando per tutti i teams, la deviazione standard del tempo di completamento è minore od uguale ad 1.0. Alla terminazione si stampino nel file `outputs.txt` i tempi ed i costi per ogni team nel seguente formato.

La prima riga del file `outputs.txt` contiene:

```
Team AvgTime AvgCost StdDevTime StdDevCost (time = xxx)
```

dove **xxx** è il valore della variabile Modelica **time** quando la simulazione viene terminata al comando **terminate**.

Le altre righe hanno il seguente formato:

<ID del Team> <Valore di AvgTime> <Valore di AvgCost> <Valore di StdDevTime> <Valore di StdDevCost>

Si usi un orizzonte di simulazione molto grande. In particolare si verifichi che l'orizzonte di simulazione sia maggiore del valore del `time` quando la simulazione è terminata dal comando `terminate`. Se questo non è verificato il modello è sbagliato. Questo valore di `time` è visibile su `stdout`.

NOTA BENE

1. Tutti i parametri del vostro modello devono essere contenuti nel record `Prm` nel file `parameters.mo`. Oltre a quelli menzionati nel testo dell'esercizio potete aggiungere dei vostri parametri, ma non dovete in alcun caso rimuovere quelli che ci sono poichè vengono usati per la correzione.
2. Il modello `System` nel file `system.mo` deve essere esteso come serve, ma non devono essere rimosso il contenuto già presente poichè viene usato per la correzione.
3. Potete aggiungere file a vostra discrezione ed estendere a vostra discrezione il contenuto dei file che vi sono forniti.
4. Salvo esplicita istruzione in senso contrario, non potete modificare in alcun modo il contenuto già presente nei file che vi sono forniti. Questi vengono usati per interfacciarsi con gli script di correzione. Una modifica delle interfacce fornite rende impossibile la correzione e quindi l'esercizio riceverà 0 punti.
5. Prima di consegnare accertarsi che il vostro modello compili. I modelli che non compilano ricevono 0 punti.
6. Prima di consegnare accertarsi che l'orizzonte di simulazione sia maggiore del valore di `time` quando il comando `terminate` termina la simulazione. Se questo non accade la simulazione è sbagliata e l'esercizio riceve 0 punti.
7. Prima di consegnare accertarsi che il file `outputs.txt` (se richiesto) sia presente e valorizzato come richiesto. Se questo file manca oppure è vuoto l'esercizio riceve 0 punti.