Chapter 7 - Quantitative Process Analysis

Flow Analysis

Cycle Time Analysis

- **Cycle Time**: the difference between a process's/task's start and end time. For a sequential fragment, it is the sum of the cycle times of the tasks in the fragment.
- **Cycle Time Analysis**: the task of calculating the **min**, **max** and **average** cycle time for an entire process or process fragment.
- However we must account for XOR splits (Alternative paths), AND splits (parallel paths) and cycles (rework).
- Cycle Time in this example is 10 + 20 = 30h.

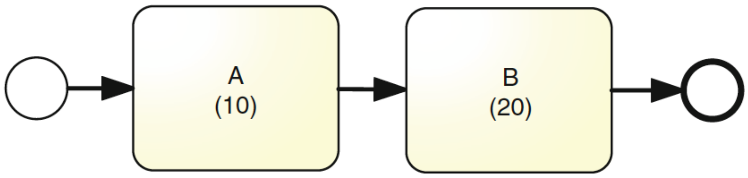


Fig. 7.1 Fully sequential process model (durations of tasks in hours are shown between brackets)

- If B is performed, the cycle time is 30h, if C is performed the cycle time is 20h, therefore the cycle time must be lower than 30h.
- However, it depends on how often each branch of the XOR-split is taken. If B is taken 50% of the times and C is taken 50% of the times then the overall cycle time is 25h.
- If the B branch is taken 90% of the times and C is taken 10% of the times, the overall cycle time is 29h.

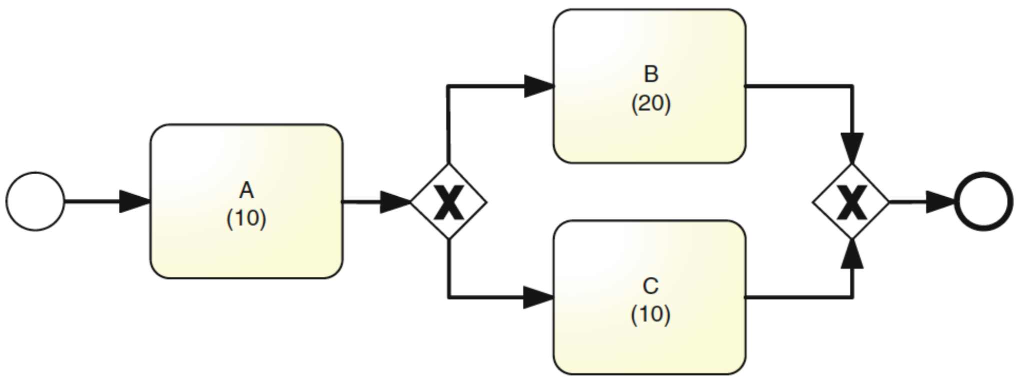
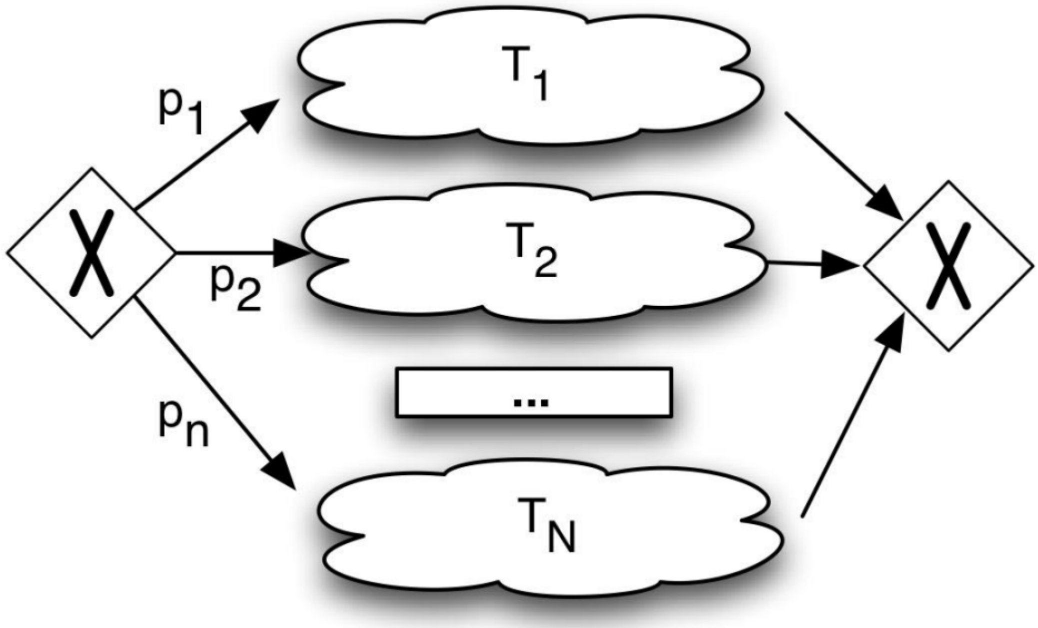


Fig. 7.2 Process model with XOR-block

- **Branch Probability**: frequency with which a given branch of a decision gateway is taken.

$$CT = \sum_{i=1}^{n} p_i \times T_i$$

Alternative Paths



$$CT = p_1T_1 + p_2T_2 + \ldots + p_nT_n = \sum_{i=1}^{n} p_i T_i$$

- Here the cycle time is determined by the slowest of the two tasks, B, therefore the cycle time is 30h.

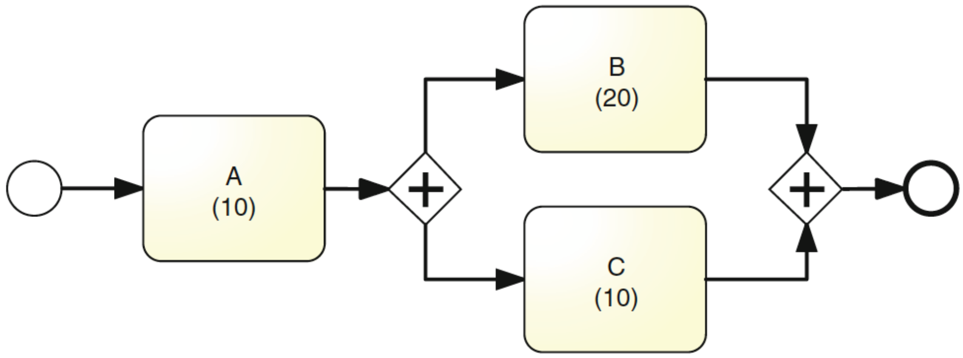
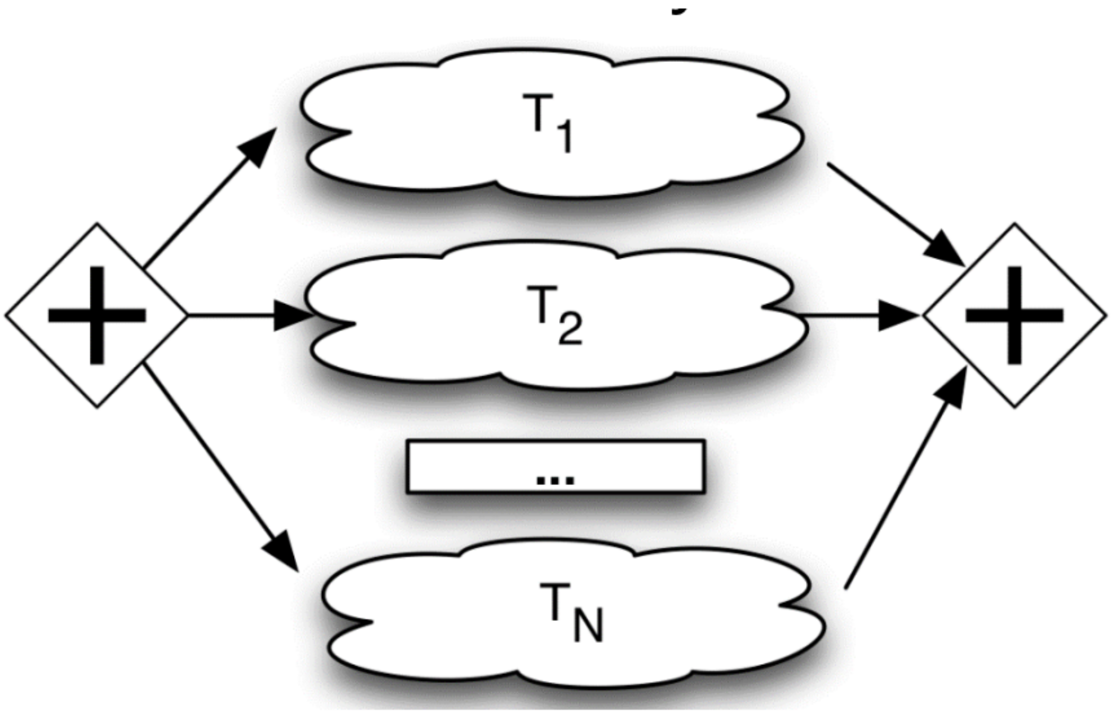


Fig. 7.4 Process model with AND-block



$$CT_{parallel} = Max\{T_1, T_2, \ldots, T_M\}$$

- Here, we know task B will be executed once, B may be repeated again with a probability of 20%, it may be repeated N times 0.2^N (this is a geometric series 1/(1-r) r = 0.2). Therefore is expected to be executed 1/(1-0.2) = 1.25 -> This number is the expected number of times B will execute. 1.25x20 + 10 = 35h.

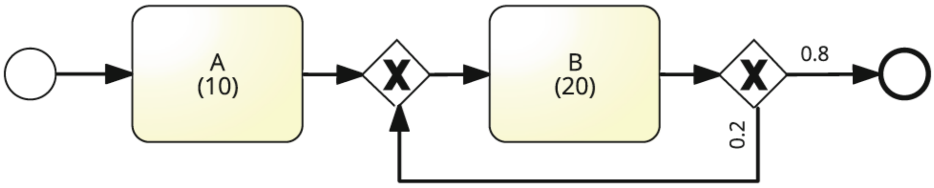


Fig. 7.7 Example of a rework block

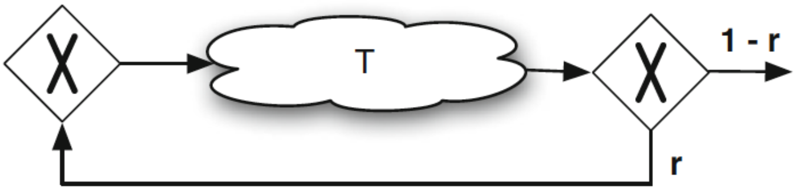


Fig. 7.8 Rework pattern

$$CT = \frac{T}{1-r}.$$

- r called **rework probability**
- In this case, 0.2 x 20 + 0.8 x 0 + 20 + 10 = 34h. The zero comes from the fact that one of the branches between the XOR split and the XOR join is empty, therefore does not contribute to cycle time.

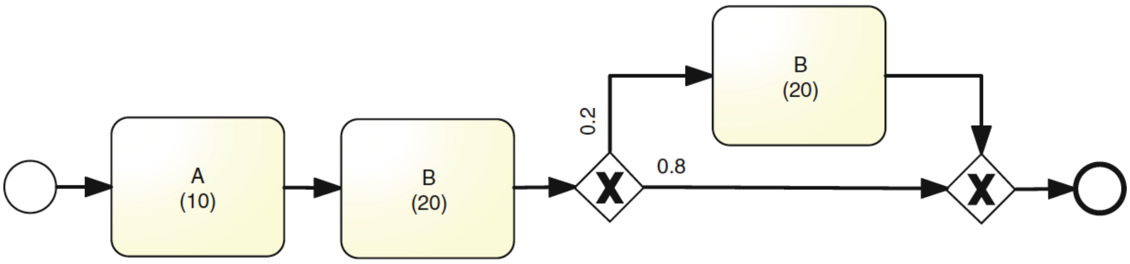
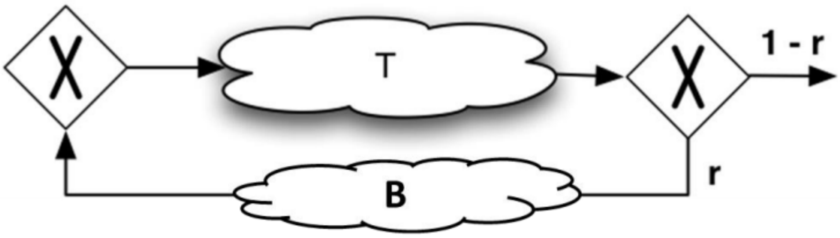


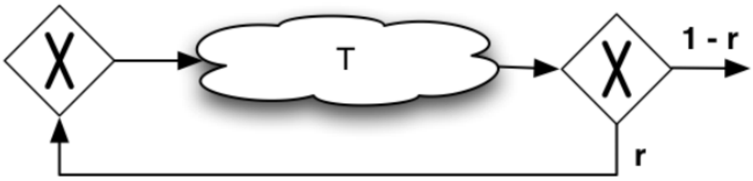
Fig. 7.9 Situation where a fragment (task) that is reworked at most once

Rework



CT = (T+B) /(1-r) - B

- *If B = 0:*



CT = T /(1-r)

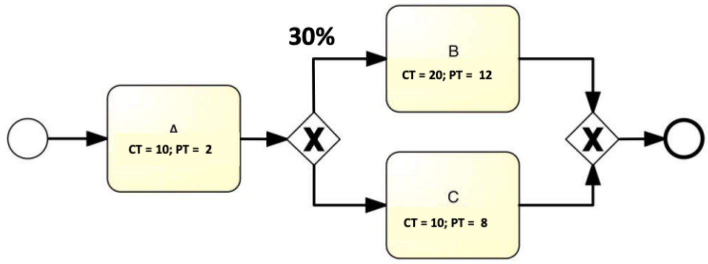
Cycle Time Efficiency

- **Cycle Time Efficiency**: measured as the percentage of the total cycle time spent on **value adding** activities.

$$\textbf{Cycle Time Efficiency} = \frac{\text{Theoretical Cycle Time}}{\text{CT}}$$

- **Theoretical Cycle Time**: it is the cycle time if we only counted **only processing times, which likely corresponds to value-adding** (including business value adding) activities and excluded any waiting time or handover time.

Alternative Paths – Example



- What is the average cycle time?

  CT = 10 + 0.3 * 20 + 0.7 *10

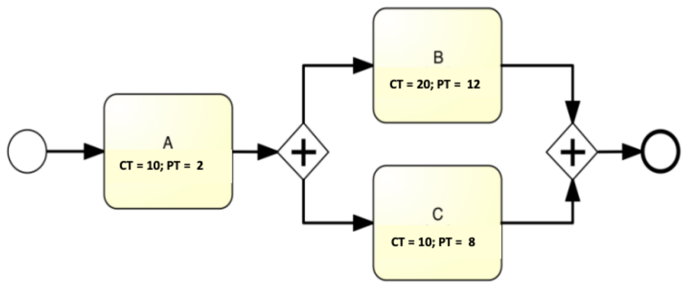
- What is the Theoretical Cycle Time

  TCT = 2 + 0.3 * 12 + 0.7 *8

- What is the average processing time (effort) ?

  Average Processing time = 2 + 0.3 * 12 + 0.7 *8

Parallel Paths – Example



- What is the average cycle time?

  CT = 10 + MAX(20 + 10)

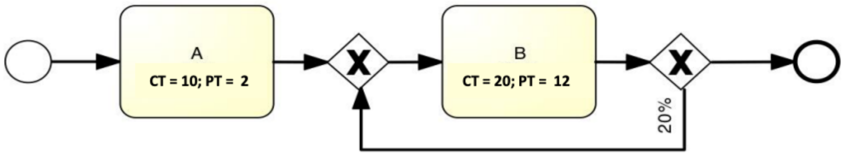
- What is the Theoretical Cycle Time

  TCT = 2 + MAX(12;8)

- What is the average processing time (effort) ?

  Average Processing time = 2 + 12 + 8

Rework – Example



- What is the average cycle time?

  CT = 10 + 20/(1-0.2)

- What is the Theoretical Cycle Time ?

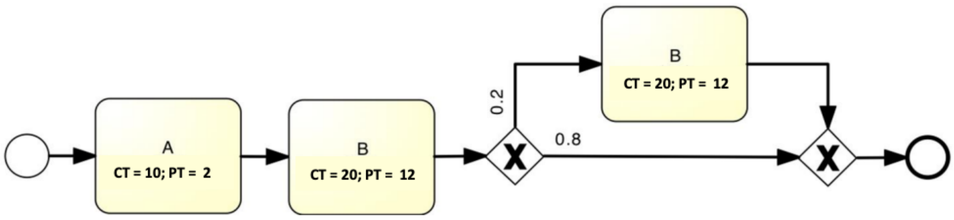
  TCT = 2 + 12/(1-0.2)

- What is the average processing time (effort) ?

  Average Processing time = 2 + 12/(1-0.2)

TECNICO
LISBOA

Rework – Example



- What is the average cycle time?

  CT = 10 + 20 + 0.2 * 20 + 0.8 * 0

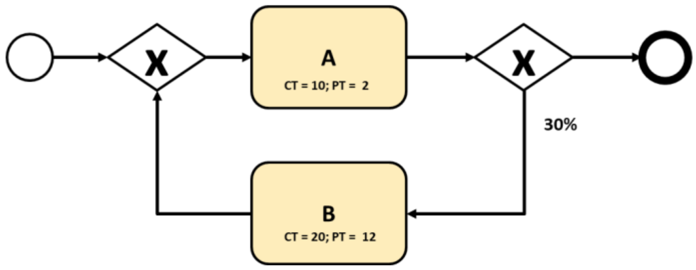
- What is the Theoretical Cycle Time

  TCT = 2 + 12 + 0.2 * 12 + 0.8 * 0

- What is the average processing time (effort) ?

  average processing time = 10 + 20 + 0.2 * 20 + 0.8 * 0

Rework – Example



- What is the average cycle time?

  CT = (10 + 20) /(1-0.3) – 20

- What is the Theoretical Cycle Time

  TCT = (2 + 12) /(1-0.3) - 12

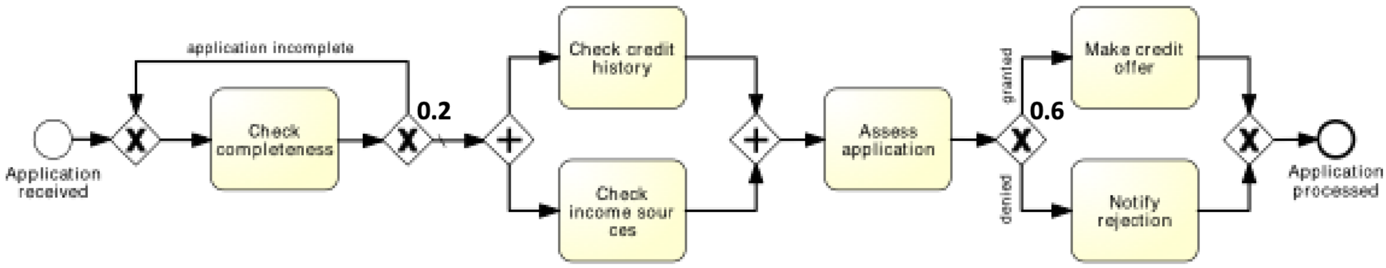
- What is the average processing time (effort) ?

  average processing time = (2 + 12) /(1-0.3) - 12

Quick Exercise

Task	Cycle time	Processing time
Check completeness	1 day	2 h
Check credit history	1 day	30 min
Check income sources	3 days	3 h
Assess application	3 days	2 h
Make credit offer	1 day	2 h
Notify rejection	2 days	30 min

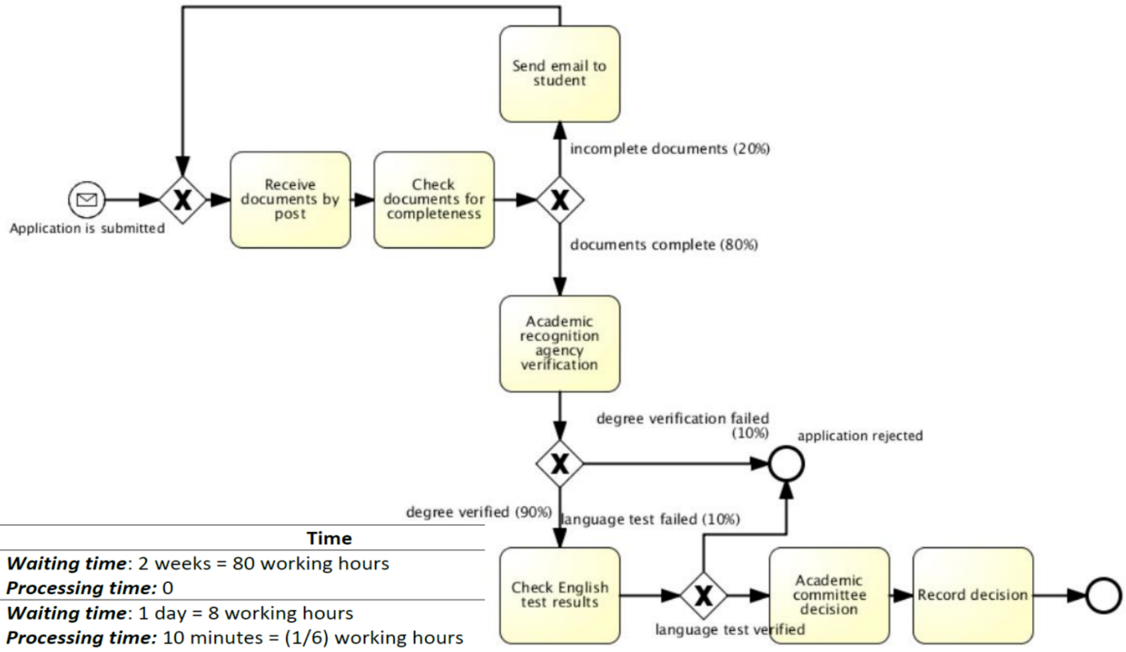
Calculate cycle time efficiency



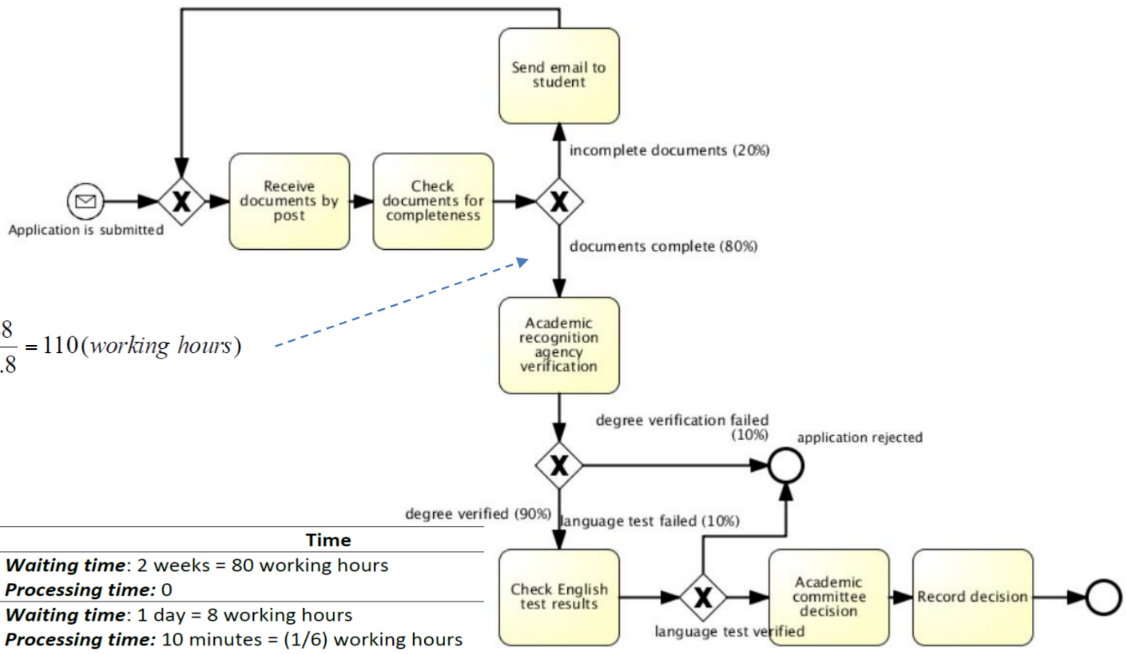
CT = 1 / (1-0.2) + MAX(1; 3) + 3 + 0.6 *1 + 0.4 * 2 = 8.65 days = 8.65*8 hrs

Theoretical Cycle Time = 2 / (1-0.2) + MAX(3; 0.5) + 2 + 0.6 *2 + 0.4 * 0.5 = 8.9 hrs

Cycle Time Efficiency = 8.9hrs /(8.65*8)hrs = 12.9%



Activity	Time
Receive documents by post	***Waiting time***: 2 weeks = 80 working hours ***Processing time:*** 0
Check documents for completeness	***Waiting time***: 1 day = 8 working hours ***Processing time:*** 10 minutes = (1/6) working hours (20% of documents are incomplete)
Academic recognition agency verification	***Waiting time***: 2 weeks = 80 working hours ***Processing time:*** 0 (10% of applications are rejected)
Check English test results	***Waiting time***: 1 day = 8 working hours ***Processing time:*** 10 minutes = (1/6) working hours (10% are rejected)
Academic committee decision	***Waiting time***: 2 weeks = 80 working hours ***Processing time:*** 1 hour
Record decision	***Waiting time***: 2 days = 16 working hours ***Processing time:*** 2 minutes = (2/60) working hours



$$CT_1 = \frac{(80+8)}{1-0.2} = \frac{88}{0.8} = 110(working\ hours)$$

Activity	Time
Receive documents by post	***Waiting time***: 2 weeks = 80 working hours ***Processing time:*** 0
Check documents for completeness	***Waiting time***: 1 day = 8 working hours ***Processing time:*** 10 minutes = (1/6) working hours (20% of documents are incomplete)
Academic recognition agency verification	***Waiting time***: 2 weeks = 80 working hours ***Processing time:*** 0 (10% of applications are rejected)
Check English test results	***Waiting time***: 1 day = 8 working hours ***Processing time:*** 10 minutes = (1/6) working hours (10% are rejected)
Academic committee decision	***Waiting time***: 2 weeks = 80 working hours ***Processing time:*** 1 hour
Record decision	***Waiting time***: 2 days = 16 working hours ***Processing time:*** 2 minutes = (2/60) working hours

CT = CT1+80+0.9*(8+0.9*(80+16)) = 274.96 working hours

17

Theoretical cycle time

The theoretical cycle time of activities is calculated by analogy to the previous calculations, but using the processing time of all activities.

$$TCT = \frac{0+\frac{1}{6}}{1-0.2} + 0 + 0.9\cdot(\frac{1}{6} + 0.9\cdot(1+\frac{2}{60})) \approx 1.2\ (working\ hours)$$

Cycle time efficiency

$$CTE = \frac{TCT}{CT} = \frac{0.54}{274.96} \approx 0.004 = 0.4\%$$

Little's Law

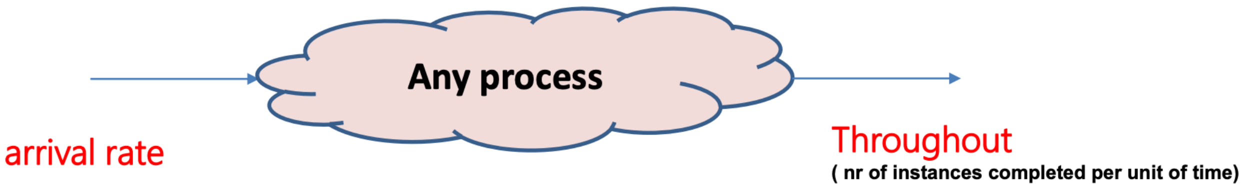
- **Arrival Rate (λ)**: average number of new instances created per time unit.
  - Ex: average # of new orders that arrive per day in an order-to-cash process.
- **Work in Progress (WIP)**: number of instances that are running (started but yet to complete).
  - Ex: # of active and unfilled orders in an order-to-cash process.

- Little's Formula: **WIP = λ·CT**
  - λ = arrival rate (number of new cases per time unit)
  - CT = cycle time (time to process each case)

Number of running instances = Arrival Rate * Time to process each instance

- This formula tells us:
  - + Arrival Rate and/or + CT means + WIP
  - If the arrival rate increases and we want to keep WIP the same, the cycle time must decrease
- **This law works only for stable processes** (meaning the number of active instances is not increasing infinitely).

Number of running instances = Arrival Rate * Time to process each instance



The system is stable when arrival rate = Throughout

- **This law can be used as an alternative to calculate Total Cycle Time of a process, all we need to know is the λ and WIP. Sometimes this is easier than determining the cycle time.**

  example, in the case of the credit application process, the arrival rate can be easily calculated if we know the total number of applications processed over a period of time. For example, if we assume there are 250 business days per year and we know the total number of credit applications over the last year is 2,500, we can infer that the average number of applications per business day is 10. WIP on the other hand can be calculated by means of sampling. We can ask how many applications are active at a given point in time, then ask this question again one week later and again two weeks later. Let us assume that on average we observe that 200 applications are active at the same time. The cycle time is then $WIP/\lambda = 200/10 = 20$ business days.

Queuing Theory

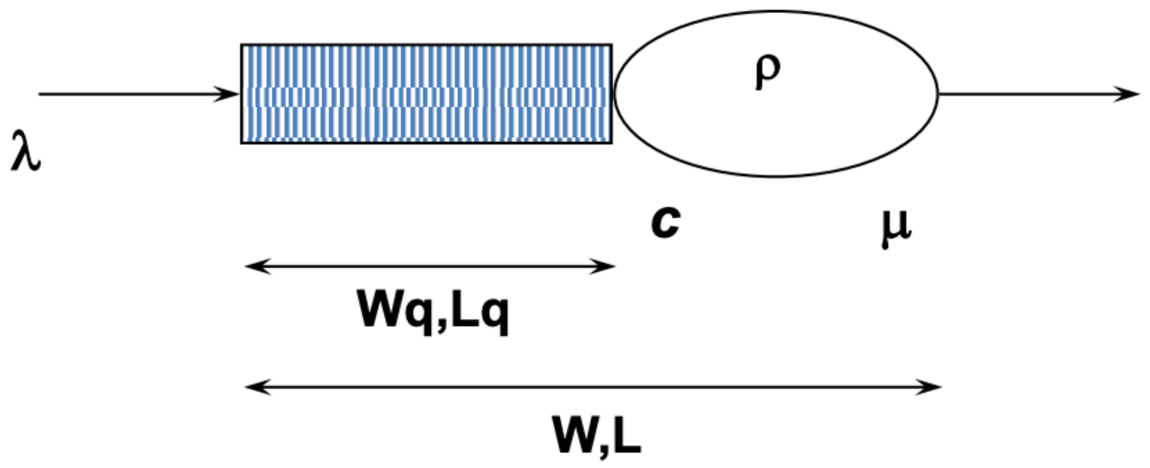
Why is Queuing Analysis important?

- Capacity problems are very common in industry and one of the main drivers of process redesign, we need to balance the cost of increased capacity against the gains of increased productivity and service.

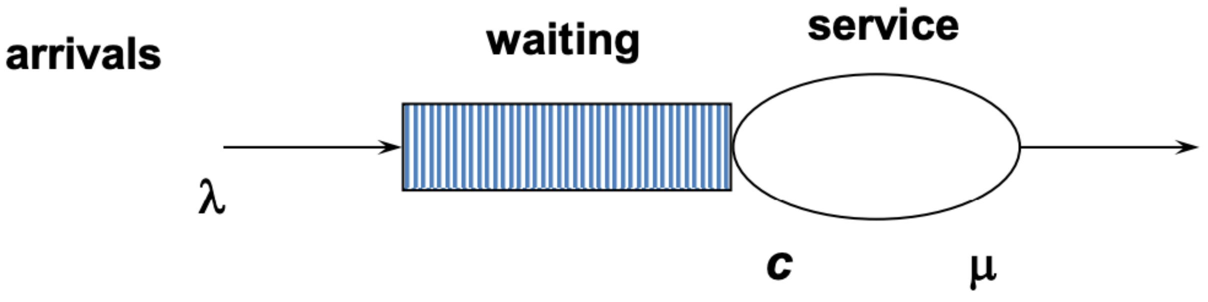
- Queuing and waiting time analysis is particularly important in service systems, large costs of waiting and of lost sales due to waiting.

Basic Concepts of Queuing theory

- Mean Arrival Rate (λ) = average number of arrivals per time unit
- Mean Service Rate (μ) = average number of jobs that can be handled by one server per time unit
- c = number of servers

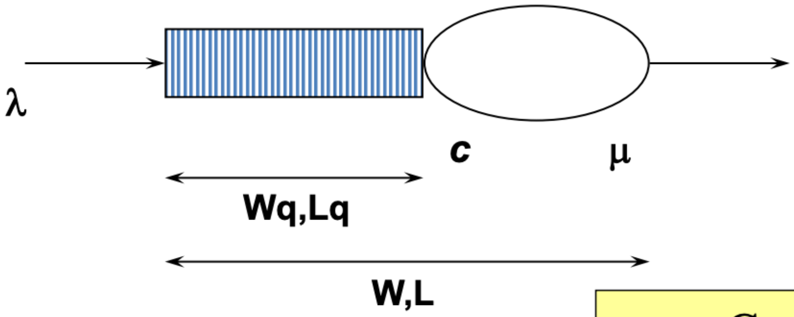


- Given λ, μ and c we can calculate:
  - ρ = occupation rate
  - Wq = average time one job spends in queue
  - Lq = average number in queue (i.e length of the queue)
  - W = average time one job spends in the system (i.e cycle time)
  - L = average number in system average (i.e WIP)



M/M/1 queue

M/M/1 queue



Assumptions:

- **time between arrivals and service time follow a negative exponential distribution**
- **1 server (*c* = 1)**
- **FIFO**

$$\rho = \frac{\text{Capacity Demand}}{\text{Available Capacity}} = \frac{\lambda}{\mu}$$

$$L = \rho/(1-\rho)$$

$$W = L/\lambda = 1/(\mu-\lambda)$$

$$L_q = \rho^2/(1-\rho)$$

$$W_q = L_q/\lambda \quad = \lambda/(\mu(\mu-\lambda))$$

M/M/c queue

- Now there are c servers in parallel so the expected capacity per time unit is then c*μ

$$\rho = \frac{\text{Capacity Demand}}{\text{Available Capacity}} = \frac{\lambda}{c * \mu}$$

Little's Formula $\Rightarrow W_q = L_q/\lambda$

$$W = W_q + (1/\mu)$$

Little's Formula $\Rightarrow L = \lambda W$

Tool Support

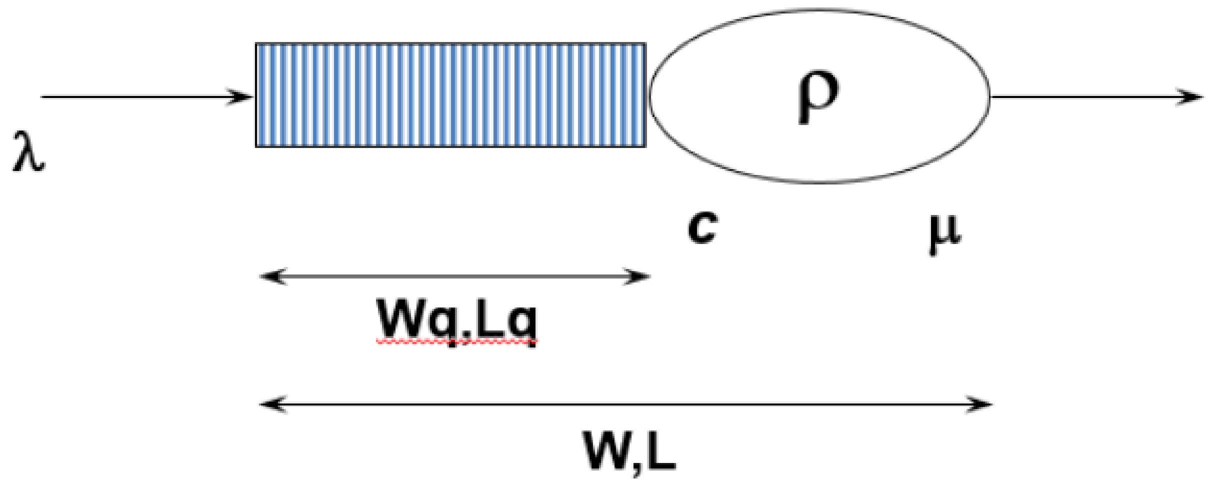
- For M/M/c the exact computation of Lq is too complex so we use a tool called rcalclite.

$$L_q = \sum_{n=c}^{\infty}(n-c)P_n = \ldots = \frac{(\lambda/\mu)^c \rho}{c!(1-\rho)^2} P_0$$

$$P_0 = \left(\sum_{n=0}^{c-1} \frac{(\lambda/\mu)^n}{n!} + \frac{(\lambda/\mu)^c}{c!} \cdot \frac{1}{1-(\lambda/(c\mu))} \right)^{-1}$$

Example - ER at County Hospital

- **Situation**
  - Patients arrive according to a Poisson process with a mean rate of λ
  - The service time (the doctor's examination and treatment time of a patient) follows an exponential distribution with mean 1/μ
  - The ER can be model as an M/M/c system where c = number of doctors
- **Data Gathering**
  - λ = 2 patients per hour (patient arrival rate)
  - μ = 3 patients per hour (patient processing rate)
- **Question**
  - Should the capacity be increased from 1 to 2 doctors?
- **Interpretation**
  - To be in queue = to be in waiting room (Wq, Lq)
  - To be in system = to be waiting or under treatment (W,L)



Characteristic	One doctor (c=1)	Two Doctors (c=2)
ρ	2/3	1/3
L_q	4/3 patients	1/12 patients
L	2 patients	3/4 patients
W_q	2/3 h = 40 minutes	1/24 h = 2.5 minutes
W	1 h	3/8 h = 22.5 minutes

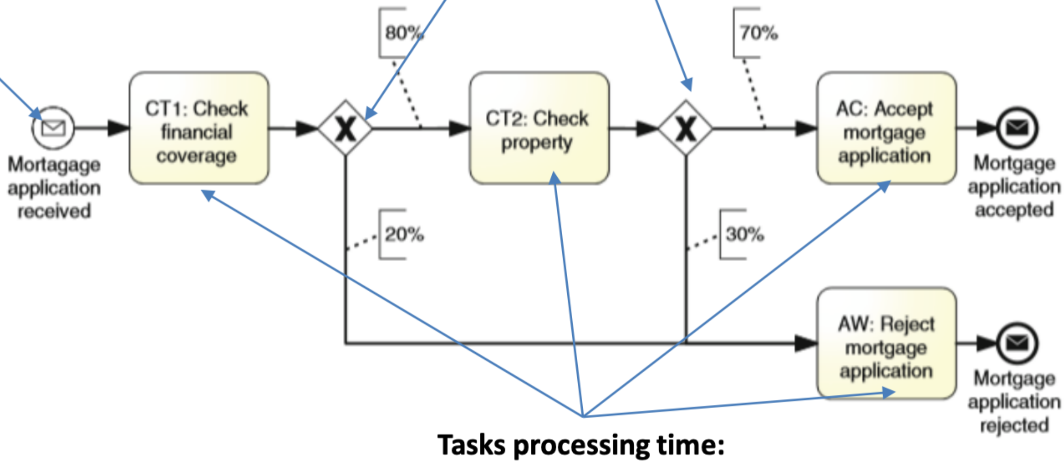
Process Simulation

- First we should present the **drawbacks of queuing theory**:
  - Generally not applicable when system includes parallel activities
  - Requires case-by-case mathematical analysis
  - Assumes steady-state (valid only for long-term analysis)
- So process simulation is **more versatile** and popular
- Process simulation = running a large number of process instances, gather data (cost, duration, resource usage) and calculate statistics from the output

Simulation Parameters

Splitting Gateways probabilities

Arrival Interval /rate:
- **Average**
- **Distribution**
- **Simulation max input events**



Tasks processing time:
- **Average**
- **Distribution**

Tasks resources
- **Shared Private pool of resources**
- **Number of Resources**
- **Working hours**