

LTAT.05.003

Software Engineering

Lecture 03/07: Estimation with Use Cases



Dietmar Pfahl

email: dietmar.pfahl@ut.ee

Fall 2019

Schedule of Lectures

Week 01: Introduction to SE

Week 02: Requirements Engineering I

Week 03: Requirements Engineering II

Week 04: Analysis

Week 05: *Development Infrastructure*

Week 06: *Continuous Development
and Integration*

Week 07: Architecture and Design I

Week 08: Architecture and Design II

Week 09: Verification and Validation I

Week 10: Verification and Validation II

Week 11: Refactoring (and TDD)

Week 12: Agile/Lean Methods

Week 13: *Agile Methods in Industry*

Week 14: Course wrap-up, review and
exam preparation

Week 15: Reserve time slot (no lecture
scheduled as of today)

Use Case Diagrams and Descriptions

Use Case Description:

Name of Use Case

Actors associated with Use Case

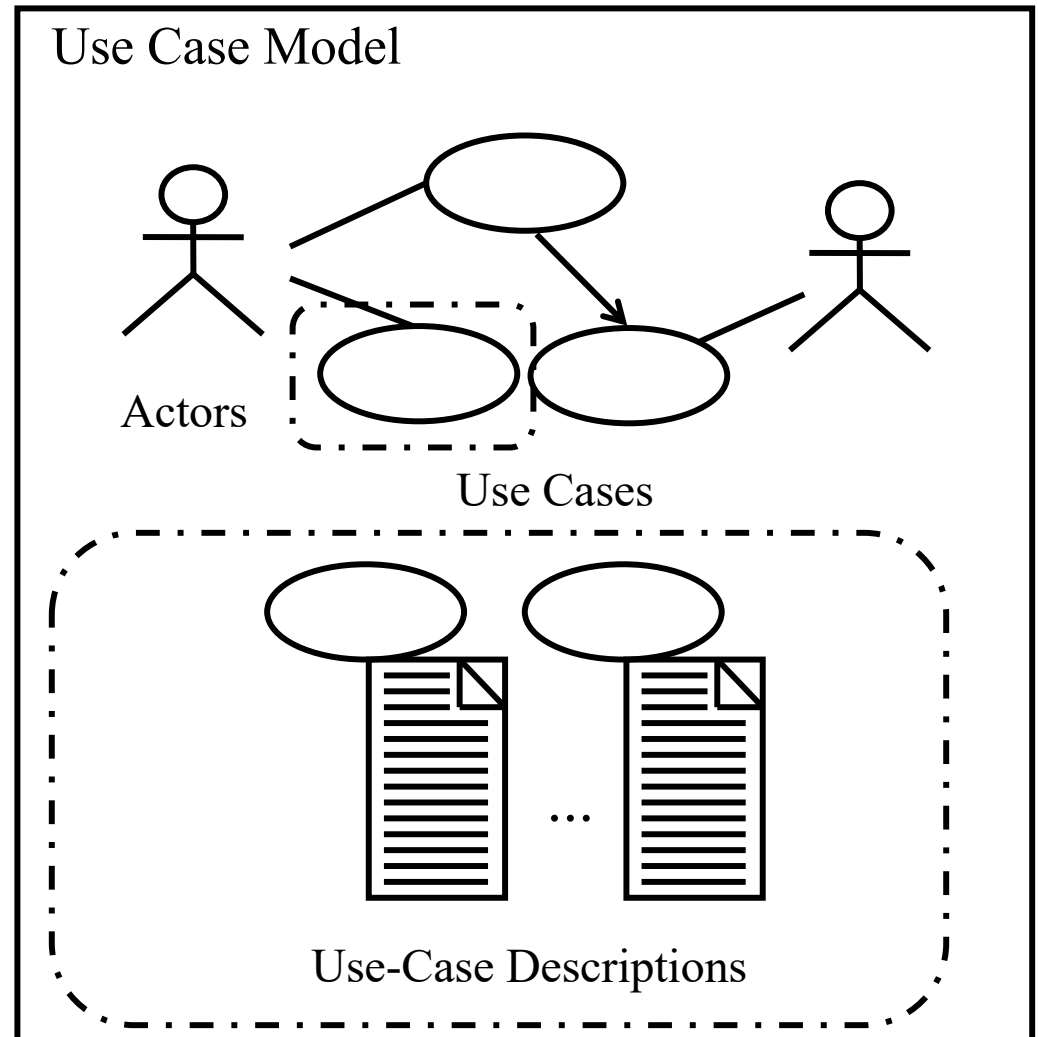
Pre-conditions

Post-conditions

Normal Flow of Events (Basic Scenario)

Alternative Flow of Events (Alternative Scenarios)

...



Use Case Diagrams and Descriptions

Use Case Description:

Name of Use Case

Actors associated with Use Case

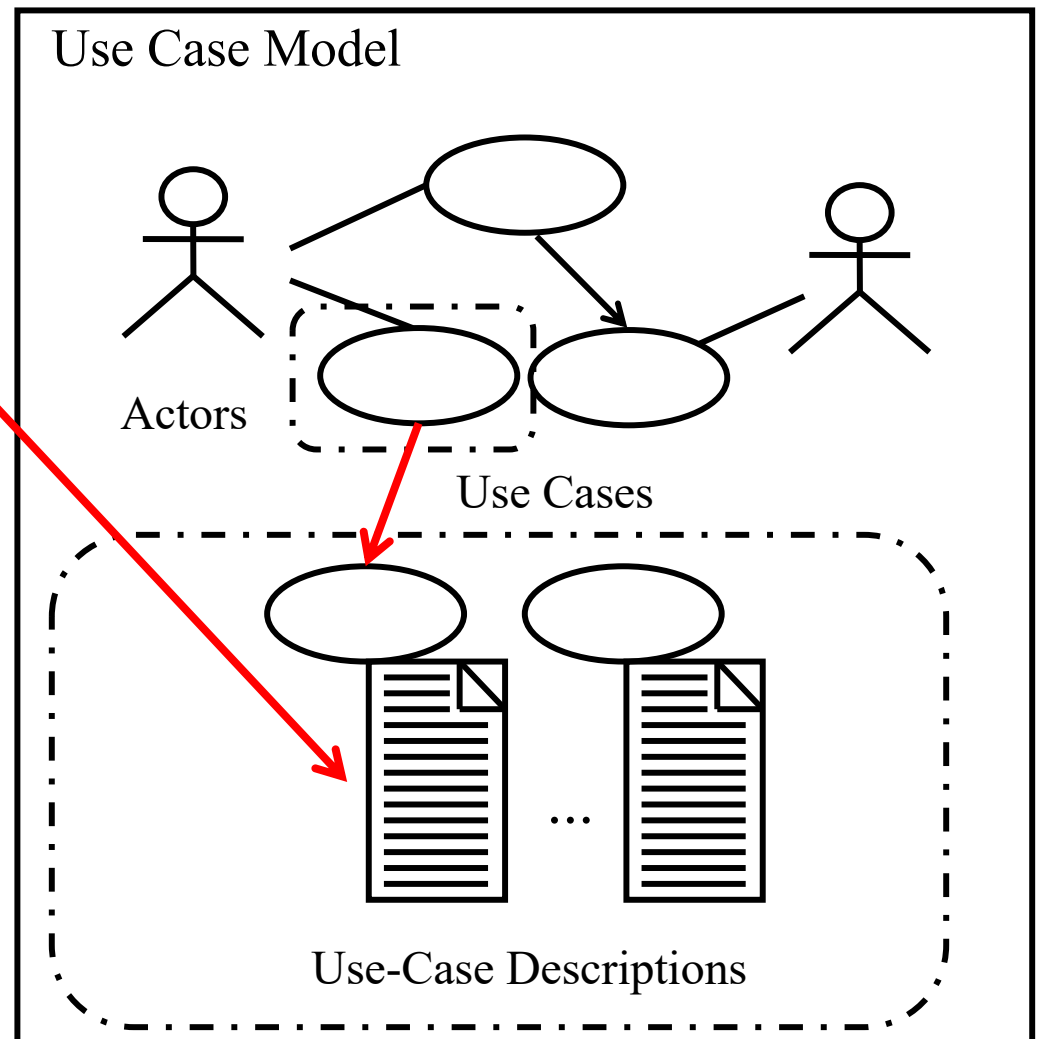
Pre-conditions

Post-conditions

Normal Flow of Events (Basic Scenario)

Alternative Flow of Events (Alternative Scenarios)

...



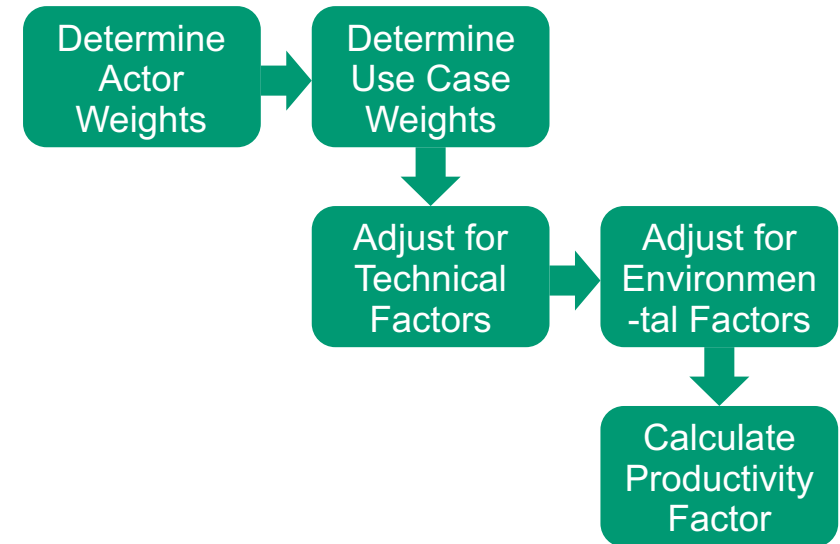
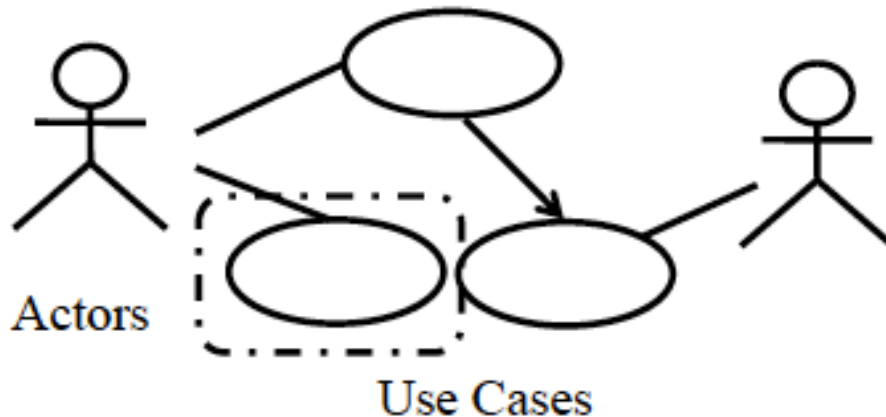
Project Estimation

with

Use Cases

Use Case Points

For all Use Cases & Actors:



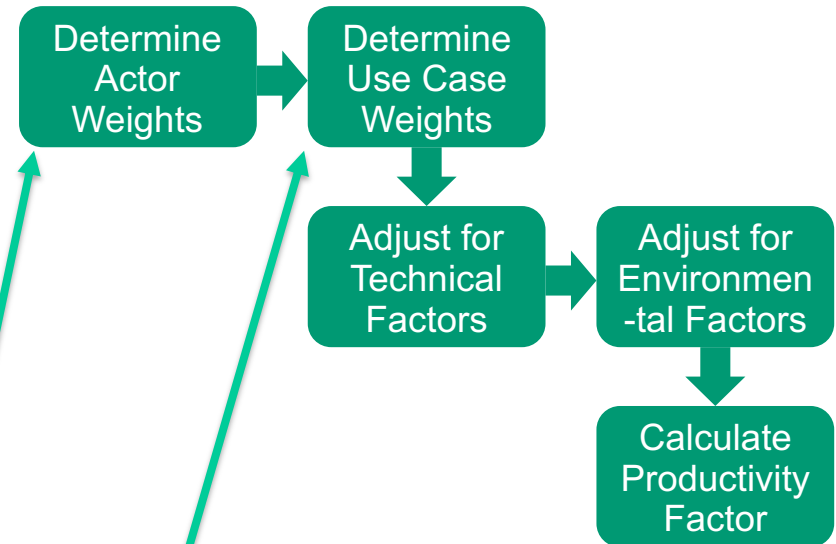
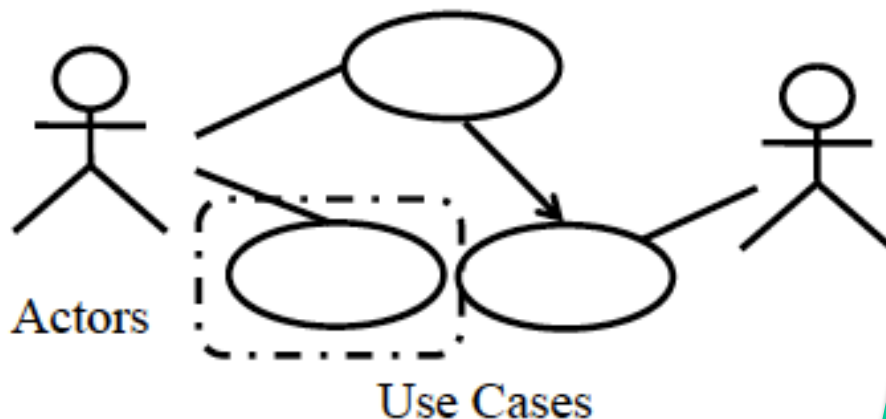
- UUCP is the sum of Unadjusted Actor Weights (UAW) and Unadjusted Use Case Weights (UUCW).

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

↑
unadjusted

Use Case Points

For all Use Cases & Actors:



- UUCP is the sum of Unadjusted Actor Weights (UAW) and Unadjusted Use Case Weights (UUCW).

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

↑
unadjusted

Unadjusted Actor Weight

Actor Type	Description	Weight
Simple	Communicates to system through API	1
Average	Interacts with the system through some protocol (HTTP, FTP, or probably some user defined protocol), or Are data stores (Files, DBMS)	2
Complex	Interacts through HCI (GUI)	3

$$\text{UAW} = (\text{Total No. of Simple actors} \times 1) + (\text{Total No. Average actors} \times 2) + (\text{Total No. Complex actors} \times 3)$$

Unadjusted Use Case Weight

Use Case Type	Description	Weight
Simple	1 to 3 transactions	5
Average	4 to 7 transactions	10
Complex	8 or more transactions	15

$$\text{UUCW} = (\text{Total No. of Simple Use Cases} \times 5) + (\text{Total No. Average Use Case} \times 10) + (\text{Total No. Complex Use Cases} \times 15)$$

Use Case Description

Use Case UC-2:	Lock
Related Requirements:	REQ1, REQ2, and REQ5 stated in Table 2-1
Initiating Actor:	Any of: Tenant, Landlord, or Timer
Actor's Goal:	To lock the door & get the lights shut automatically (?)
Participating Actors:	LockDevice, LightSwitch, Timer
Preconditions:	The system always displays the menu of available functions.
Postconditions:	The door is closed and lock armed & the auto-lock timer is reset.
Flow of Events for Main Success Scenario:	
→	1. Tenant/Landlord selects the menu item "Lock"
←	2. System (a) signals affirmation, e.g., "lock armed," (b) signals to LockDevice to arm the lock (if not already armed), (c) signal to Timer to reset the auto-lock counter, and (d) signals to LightSwitch to turn the light off (?)
Flow of Events for Extensions (Alternate Scenarios):	
2a. System senses that the door is not closed, so the lock cannot be armed	
←	1. System (a) signals a warning that the door is open, and (b) signal to Timer to start the alarm counter
→	2. Tenant/Landlord closes the door
←	3. System (a) senses the closure, (b) signals affirmation to the Tenant/Landlord , (c) signals to LockDevice to arm the lock, (d) signal to Timer to reset the auto-lock counter, and (e) signal to Timer to reset the alarm counter

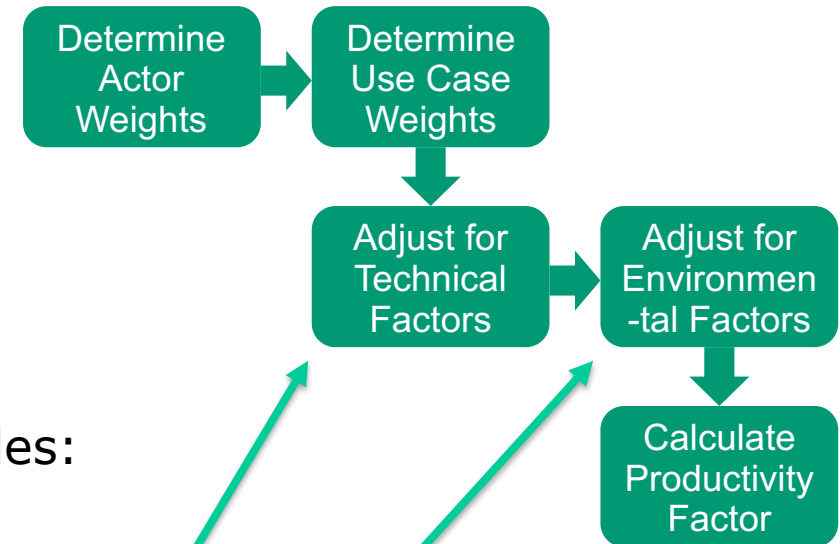
Transactions



Use Case Points

For all Use Cases & Actors:

- UCP equation is composed of three variables:
 - Unadjusted Use Case Point (UUCP)
 - The Technical Complexity Factor (TCF)
 - The Environment Complexity Factor (ECF)



$$\text{(adjusted) UCP} = \text{UUCP} * \text{TCF} * \text{ECF}$$

- UUCP is the sum of Unadjusted Actor Weights (UAW) and Unadjusted Use Case Weights (UUCW).

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

↑
unadjusted

Technical Complexity Factor – TCF

Technical Factor	Description	Weight
TF(1)	Distributed System	2
TF(2)	Performance	1
TF(3)	End User Efficiency	1
TF(4)	Complex Internal Processing	1
TF(5)	Reusability	1
TF(6)	Installability	0.5
TF(7)	Usability	0.5
TF(8)	Portability	2
TF(9)	Modifiability	1
TF(10)	Concurrency	1
TF(11)	Includes special security requirements	1
TF(12)	Provides direct access by third parties	1
TF(13)	Special User training facilities are required	1

Each TF(i) can have a value from 0 (factor is irrelevant) to 5 (factor is essential)

$$TCF = 0.6 + TF/100 \text{ with } TF = \sum_{i=1}^{13} (TF(i) * Weight(i))$$

Environmental Complexity Factor - ECF

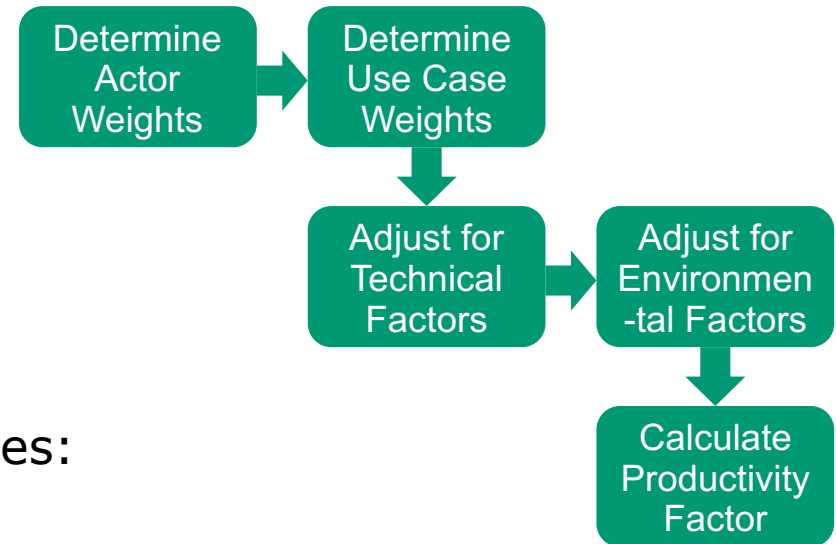
Factor Number	Description	Weight
EF(1)	Familiarity with system development process in use	1.5
EF(2)	Application experience	0.5
EF(3)	Object-oriented experience	1.0
EF(4)	Lead analyst capability	0.5
EF(5)	Motivation	1.0
EF(6)	Requirements stability	2.0
EF(7)	Part time staff	-1.0
EF(8)	Difficulty of programming language	-1.0

Each EF(i) can have a value from 0 (no experience) to 5 (expert)

Complexity increases, the smaller EF(1) to EF(6) and the greater EF(7) & EF(8)

$$ECF = 1.4 + (-0.03 * EF) \text{ with } EF = \sum_{i=1}^8 (EF(i) * Weight(i))$$

Use Case Points



For all Use Cases & Actors:

- UCP equation is composed of three variables:
 - Unadjusted Use Case Point (UUCP)
 - The Technical Complexity Factor (TCF)
 - The Environment Complexity Factor (ECF)

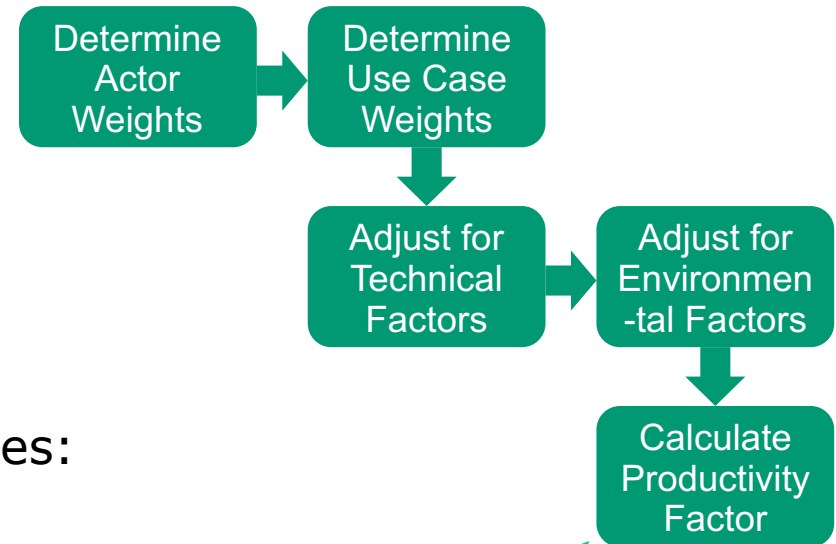
Effort Estimation ?

$$\mathbf{UCP = UUCP * TCF * ECF}$$

- UUCP is the sum of Unadjusted Actor Weights (UAW) and Unadjusted Use Case Weights (UUCW).

$$\mathbf{UUCP = UAW + UUCW}$$

Use Case Points



For all Use Cases & Actors:

- UCP equation is composed of three variables:
 - Unadjusted Use Case Point (UUCP)
 - The Technical Complexity Factor (TCF)
 - The Environment Complexity Factor (ECF)

- Effort Estimate uses the Productivity Factor (PF):

$$\text{Effort(UCP)} = \text{UUCP} * \text{TCF} * \text{ECF} * \text{PF}$$

- UUCP is the sum of Unadjusted Actor Weights (UAW) and Unadjusted Use Case Weights (UUCW).

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

Use Case Points: Project Effort

For all Use Cases:

- UCP equation is composed of three variables:
 - Unadjusted Use Case Point (UUCP)
 - The Technical Complexity Factor (TCF)
 - The Environment Complexity Factor (ECF)

Either
20 person-hours/UCP
or
28 person-hours/UCP

- Effort Estimate uses the Productivity Factor:

$$\text{Effort(UCP)} = \text{UUCP} * \text{TCF} * \text{ECF} * \text{PF [person-hours]}$$

- UUCP is the sum of Unadjusted Actor Weights (UAW) and Unadjusted Use Case Weights (UUCW).

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

Productivity Factor - PF

Complexity increases,
the smaller EF(1) to
EF(6) and the greater
EF(7) & EF(8)

If sum of [(number of factors E1 through E6 assigned value < 3)
and (number of factors E7 and E8 assigned value > 3)] ≤ 2

$$PF = 20 \text{ ph/UCP}$$

EF values:
0, 1, 2, 3, 4, 5

Else

If sum of [(number of factors E1 through E6 assigned value < 3)
and (number of factors E7 and E8 assigned value > 3)] = 3 or 4

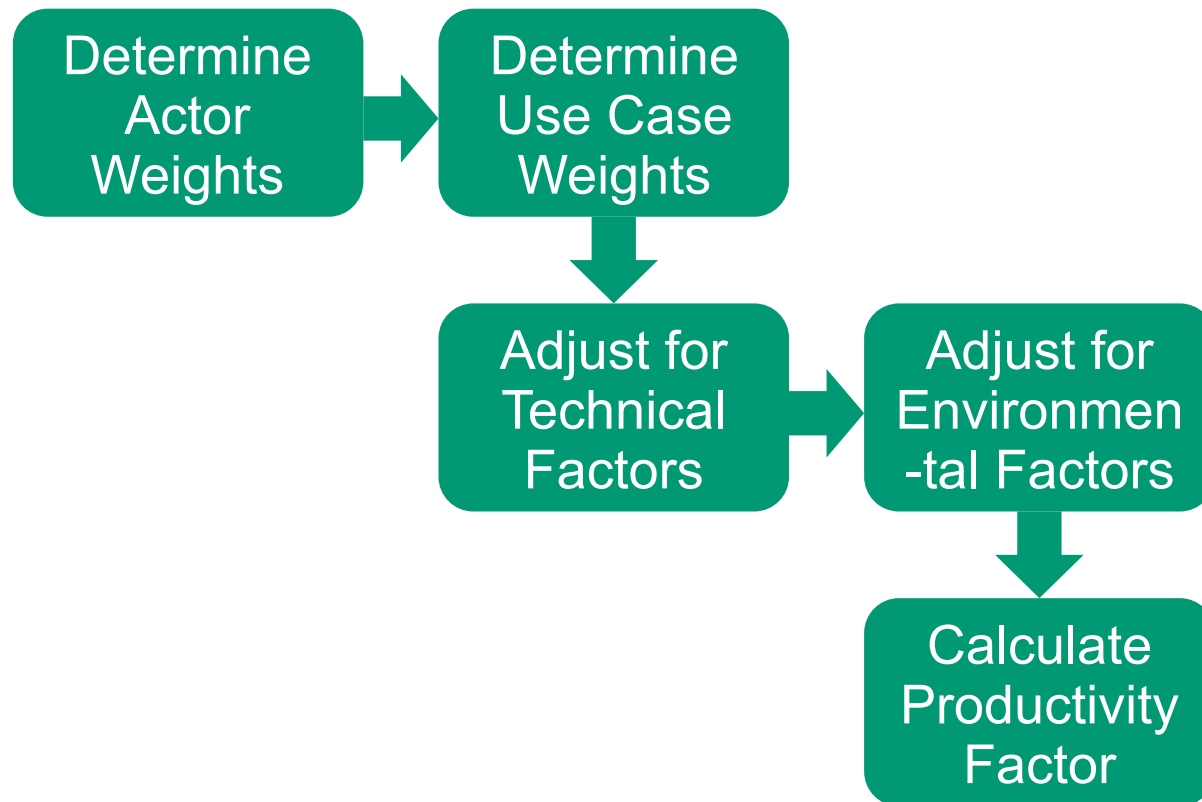
$$PF = 28 \text{ ph/UCP}$$

Else: **Rethink project; it has too high a risk of failure**

Example: http://en.wikipedia.org/wiki/Use_Case_Points

Use Case Point Calculation and Effort Estimation Example

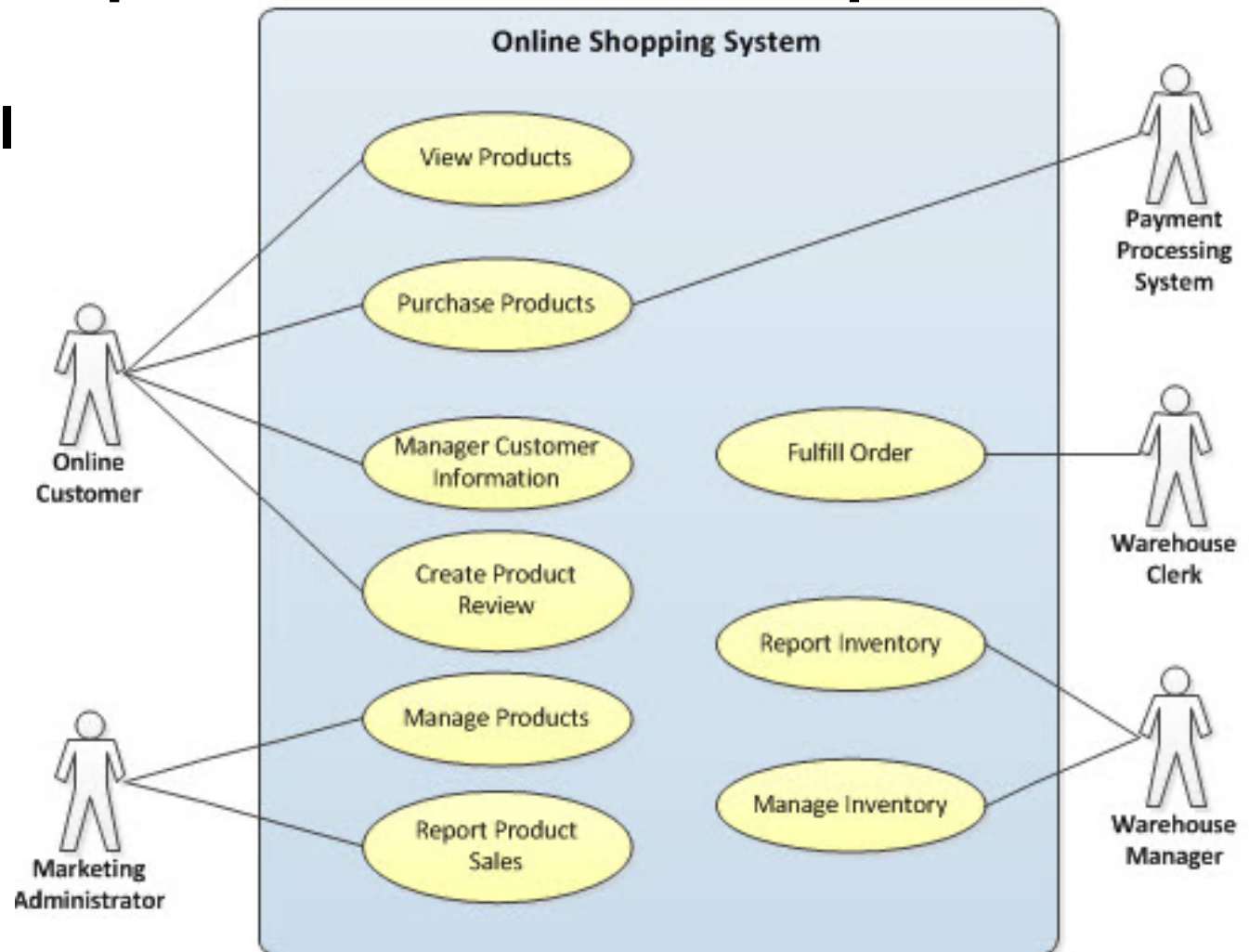
5 Steps to perform ...



UCP-Example: Online Shop

Use Case Model

- 5 Actors
- 9 UCs



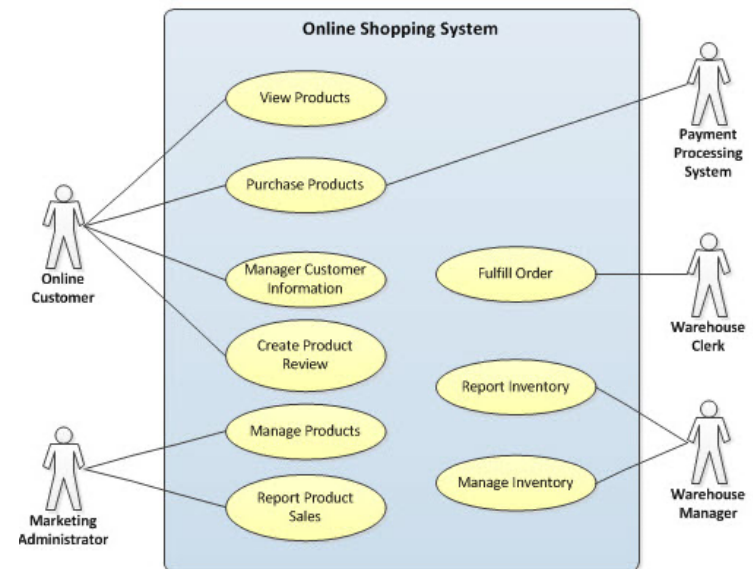
Example Source: Wikipedia (Oct 2019)

1. Determine Actor Weights (UAW)

What do we know?

The UC model shows five actors:

- Payment Processing System
- Online Customer
- Marketing Administrator
- Warehouse Clerk
- Warehouse Manager



Which actors are simple, average, complex?

Recall: Unadjusted Actor Weight

Actor Type	Description	Weight
Simple	Communicates to system through API	1
Average	Interacts with the system through some protocol (HTTP, FTP, or probably some user defined protocol), or Are data stores (Files, DBMS)	2
Complex	Interacts through HCI (GUI)	3

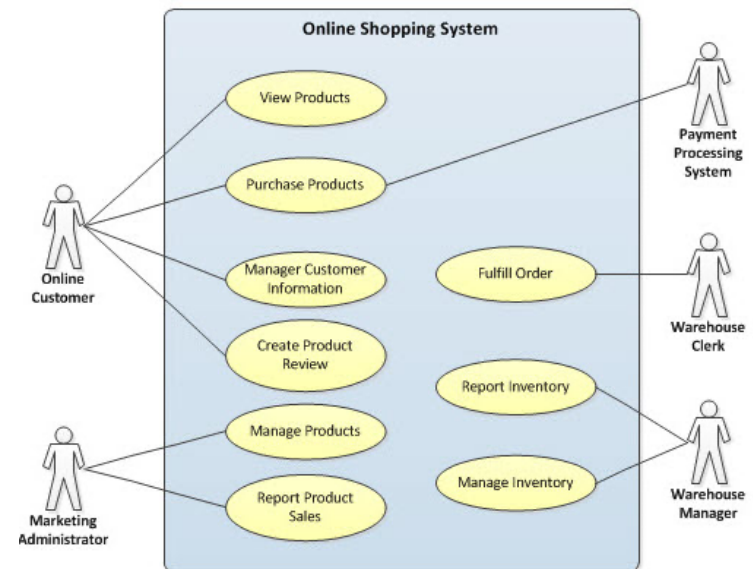
$$\text{UAW} = (\text{Total No. of Simple actors} \times 1) + (\text{Total No. Average actors} \times 2) + (\text{Total No. Complex actors} \times 3)$$

1. Determine Actor Weights (UAW)

What do we know?

The UC model shows five actors:

- Payment Processing System -> S
- Online Customer -> C
- Marketing Administrator -> C
- Warehouse Clerk -> C
- Warehouse Manager -> C



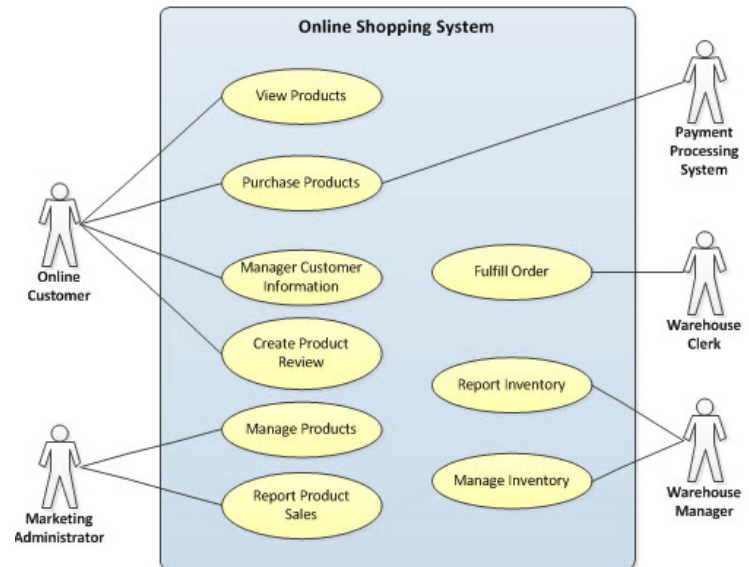
$$\text{UAW} = 1 + 3 + 3 + 3 + 3 = 13$$

2. Determine UC Weights (UUCW)

What do we know?

The UC model shows nine UCs:

- View Products
- Purchase Products
- Manage Customer Information
- Create Product Review
- Fulfill Order
- Report Inventory
- Manage Inventory
- Manage Products
- Report Product Sales



Which UCs are simple, average, complex?

Recall: Unadjusted UC Weight

Use Case Type	Description	Weight
Simple	1 to 3 transactions	5
Average	4 to 7 transactions	10
Complex	8 or more transactions	15

$$\text{UUCW} = (\text{Total No. of Simple Use Cases} \times 5) + (\text{Total No. Average Use Case} \times 10) + (\text{Total No. Complex Use Cases} \times 15)$$

2. Determine UC Weights (UUCW)

Use Case UC-2: Lock	
Related Requirements:	REQ1, REQ2, and REQ5 stated in Table 2-1
Initiating Actor:	Any of: Tenant, Landlord, or Timer
Actor's Goal:	To lock the door & get the lights shut automatically (?)
Participating Actors:	LockDevice, LightSwitch, Timer
Preconditions:	The system always displays the menu of available functions.
Postconditions:	The door is closed and lock armed & the auto-lock timer is reset.
Flow of Events for Main Success Scenario:	
→	1. Tenant/Landlord selects the menu item "Lock"
	System (a) signals affirmation, e.g., "lock armed," (b) signals to LockDevice to arm the lock (if not
←	2. already armed), (c) signal to Timer to reset the auto-lock counter, and (d) signals to LightSwitch to turn the light off (?)
Flow of Events for Extensions (Alternate Scenarios):	
2a. System senses that the door is not closed, so the lock cannot be armed	
←	1. System (a) signals a warning that the door is open, and (b) signal to Timer to start the alarm counter
→	2. Tenant/Landlord closes the door
	System (a) senses the closure, (b) signals affirmation to the Tenant/Landlord , (c) signals to LockDevice
←	3. to arm the lock, (d) signal to Timer to reset the auto-lock counter, and (e) signal to Timer to reset the alarm counter

UC Example

Simple, average or complex?

2. Determine UC Weights (UUCW)

Use Case UC-2:	Lock	UC Example
Related Requirements:	REQ1, REQ2, and REQ5 stated in Table 2-1	
Initiating Actor:	Any of: Tenant, Landlord, or Timer	
Actor's Goal:	To lock the door & get the lights shut automatically (?)	
Participating Actors:	LockDevice, LightSwitch, Timer	
Preconditions:	The system always displays the menu of available functions.	
Postconditions:	The door is closed and lock armed & the auto-lock timer is reset.	
Flow of Events for Main Success Scenario:		
→	1. Tenant/Landlord selects the menu item "Lock"	
	System (a) signals affirmation, e.g., "lock armed," (b) signals to LockDevice to arm the lock (if not	
←	2. already armed), (c) signal to Timer to reset the auto-lock counter, and (d) signals to LightSwitch to turn the light off (?)	
Flow of Events for Extensions (Alternate Scenarios):		
2a. System senses that the door is not closed, so the lock cannot be armed		
←	1. System (a) signals a warning that the door is open, and (b) signal to Timer to start the alarm counter	
→	2. Tenant/Landlord closes the door	
	System (a) senses the closure, (b) signals affirmation to the Tenant/Landlord , (c) signals to LockDevice	
←	3. to arm the lock, (d) signal to Timer to reset the auto-lock counter, and (e) signal to Timer to reset the alarm counter	

No. of Transactions = 5 → UC is average

2. Determine UC Weights (UUCW)

Use Case UC-2: Lock	
Related Requirements:	REQ1, REQ2, and REQ5 stated in Table 2-1
Initiating Actor:	Any of: Tenant, Landlord, or Timer
Actor's Goal:	To lock the door & get the lights shut automatically (?)
Participating Actors:	LockDevice, LightSwitch, Timer
Preconditions:	The system always displays the menu of available functions.
Postconditions:	The door is closed and lock armed & the auto-lock timer is reset.
Flow of Events for Main Success Scenario:	
→	1. Tenant/Landlord selects the menu item "Lock"
	System (a) signals affirmation, e.g., "lock armed," (b) signals to LockDevice to arm the lock (if not
←	2. already armed), (c) signal to Timer to reset the auto-lock counter, and (d) signals to LightSwitch to turn the light off (?)
Flow of Events for Extensions (Alternate Scenarios):	
2a. System senses that the door is not closed, so the lock cannot be armed	
←	1. System (a) signals a warning that the door is open, and (b) signal to Timer to start the alarm counter
→	2. Tenant/Landlord closes the door
	System (a) senses the closure, (b) signals affirmation to the Tenant/Landlord , (c) signals to LockDevice
←	3. to arm the lock, (d) signal to Timer to reset the auto-lock counter, and (e) signal to Timer to reset the alarm counter

UC Example

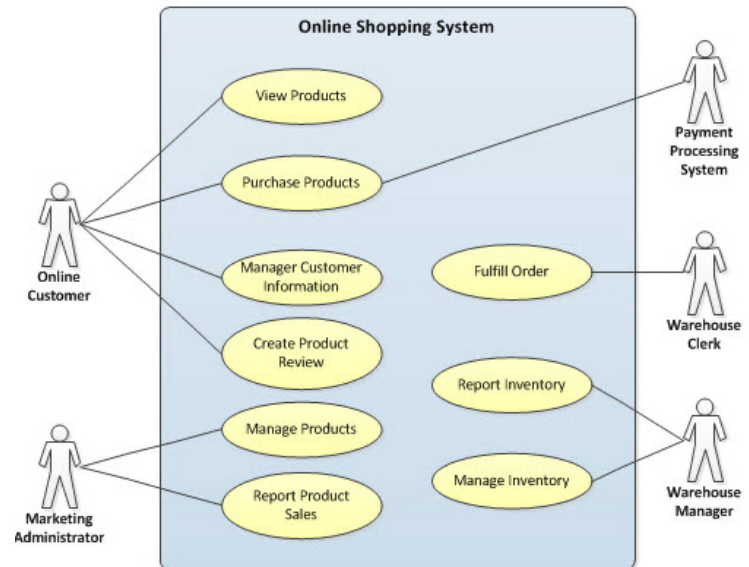
UC is average → UUCW = 10

2. Determine UC Weights (UUCW)

What do we know?

Assume we have:

- 2 simple UCs
- 3 average UCs
- 4 complex UCs

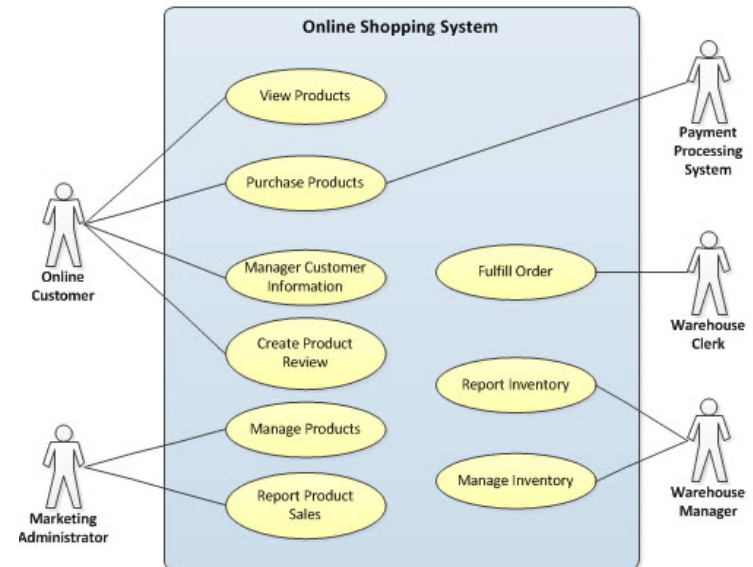


$$\text{UUCW} = 2 \times 5 + 3 \times 10 + 4 \times 15 = 100$$

3. Adjust for Technical Factors (TCF)

How do we calculate TCF?

- Each of the 13 technical factors is assigned a value based on how essential the technical aspect is to the system being developed.
 - 5: factor is essential
 - ...
 - 0: factor is irrelevant
- The values are multiplied by the corresponding weights and the total TCF is determined by:

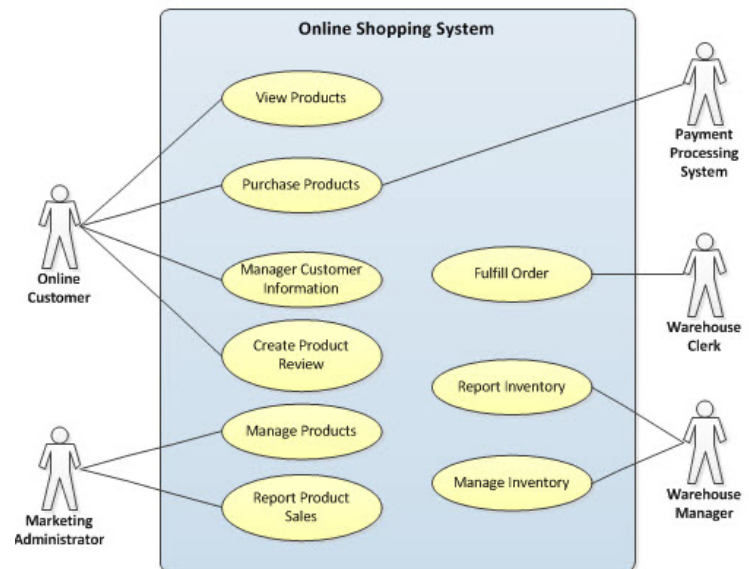


$$TCF = 0.6 + TF/100 \text{ with } TF = \sum_{i=1}^{13} (TF(i) * Weight(i))$$

3. Adjust for Technical Factors (TCF)

Let's assume:

1. Distributed System -> ?
2. Performance -> ?
3. End User Efficiency ->
4. Complex Internal Processing -> ?
5. Reusability ->
6. Installability ->
7. Usability -> ?
8. Portability ->
9. Modifiability -> ?
10. Concurrency ->
11. Includes special security requirements ->
12. Provides direct access by third parties ->
13. Special User training facilities are required -> ?

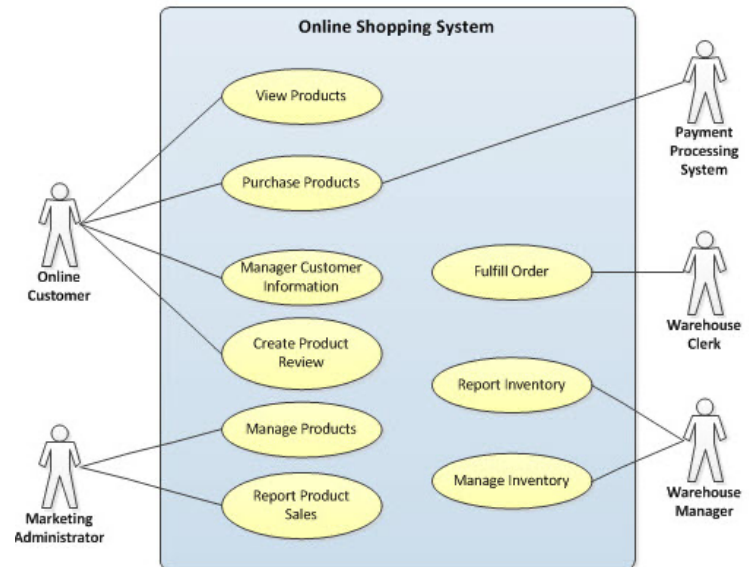


• TCF = ?

3. Adjust for Technical Factors (TCF)

Let's assume:

1. Distributed System -> **5**
2. Performance -> **5**
3. End User Efficiency -> **3**
4. Complex Internal Processing -> **2**
5. Reusability -> **3**
6. Installability -> **1**
7. Usability -> **5**
8. Portability -> **2**
9. Modifiability -> **2**
10. Concurrency -> **3**
11. Includes special security requirements -> **5**
12. Provides direct access by third parties -> **1**
13. Special User training facilities are required -> **1**



• **TCF = ?**

3. Adjust for Technical Factors (TCF)

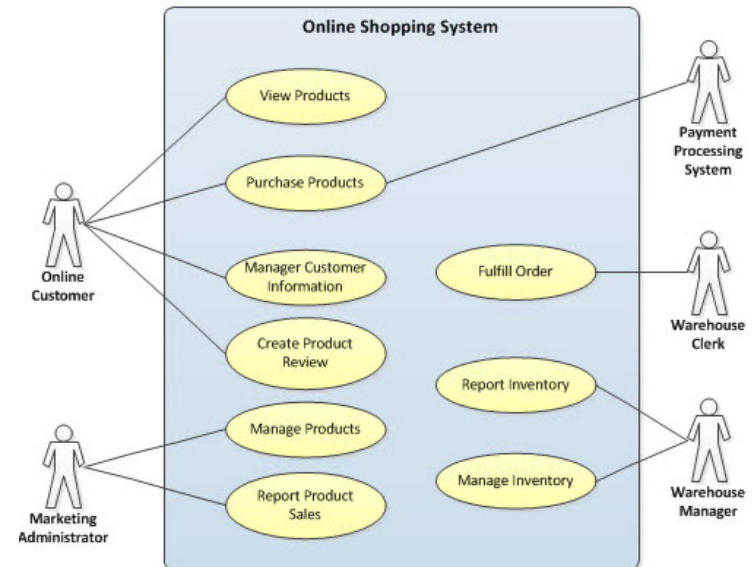
Factor	Description	Weight	Assigned Value	Weight x Assigned Value
T1	Distributed system	2.0	5	10
T2	Response time/performance objectives	1.0	5	5
T3	End-user efficiency	1.0	3	3
T4	Internal processing complexity	1.0	2	2
T5	Code reusability	1.0	3	3
T6	Easy to install	0.5	1	0.5
T7	Easy to use	0.5	5	2.5
T8	Portability to other platforms	2.0	2	4
T9	System maintenance	1.0	2	2
T10	Concurrent/parallel processing	1.0	3	3
T11	Security features	1.0	5	5
T12	Access for third parties	1.0	1	1
T13	End user training	1.0	1	1
				42

$TCF = 0.6 + 42/100 = 1.02$

4. Adjust for Environmental Factors (ECF)

How do we calculate ECF?

- Each each of the 8 environmental factors is assigned a value based on the team experience level (or an appropriate equivalent).
 - 5: team experience is expert level
 - ...
 - 0: team has no experience
- The values are multiplied by the corresponding weights and the total ECF is determined by:

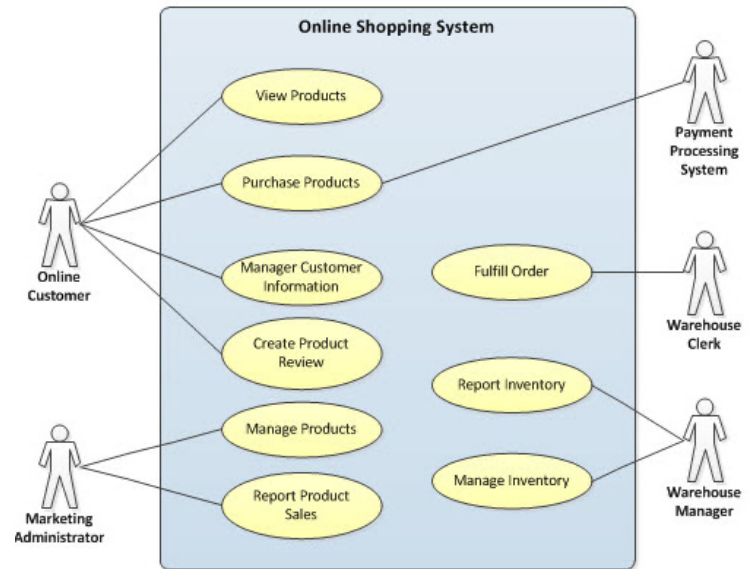


$$ECF = 1.4 + (-0.03 * EF) \text{ with } EF = \sum_{i=1}^8 (EF(i) * Weight(i))$$

4. Adjust for Environmental Factors (ECF)

Let's assume:

1. Familiarity with system development process in use -> **3**
2. Application experience -> **3**
3. Object-oriented experience -> **2**
4. Lead analyst capability -> **5**
5. Motivation -> **2**
6. Requirements stability -> **1**
7. Part time staff -> **0**
8. Difficulty of programming language -> **4**



- **ECF = ?**

4. Adjust for Environmental Factors (ECF)

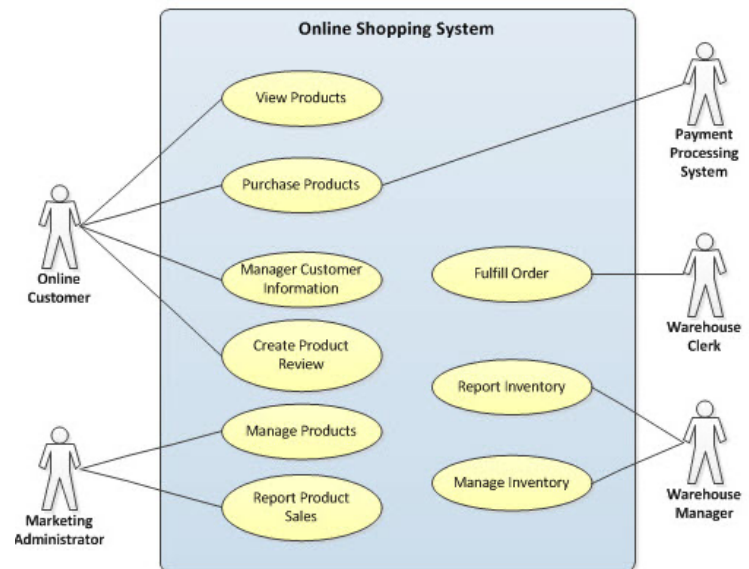
Factor	Description	Weight	Assigned Value	Weight x Assigned Value
E1	Familiarity with development process used	1.5	3	4.5
E2	Application experience	0.5	3	1.5
E3	Object-oriented experience of team	1.0	2	2
E4	Lead analyst capability	0.5	5	2.5
E5	Motivation of the team	1.0	2	2
E6	Stability of requirements	2.0	1	2
E7	Part-time staff	-1.0	0	0
E8	Difficult programming language	-1.0	4	-4
Total (EF):				10.5

$$ECF = 1.4 + (-0.03 * 10.5) = 1.085$$


5. Calculate Productivity Factor (PF)

How do we calculate PF?

- If sum of [(number of factors E1 through E6 assigned value < 3) and (number of factors E7 and E8 assigned value > 3)] ≤ 2
 - PF = **20** ph/UCP
- Else: If sum of [(number of factors E1 through E6 assigned value < 3) and (number of factors E7 and E8 assigned value > 3)] = 3 or 4
 - PF = **28** ph/UCP
- Else: Rethink project; it is too risky



Complexity increases, the smaller EF(1) to EF(6) and the greater EF(7) & EF(8)

5. Calculate Productivity Factor (PF)

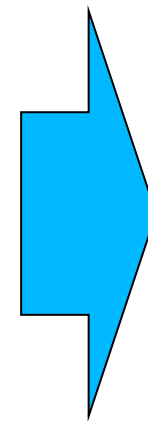
Factor	Description	Weight	Assigned Value
E1	Familiarity with development process used	1.5	3
E2	Application experience	0.5	3
E3	Object-oriented experience of team	1.0	2
E4	Lead analyst capability	0.5	5
E5	Motivation of the team	1.0	2
E6	Stability of requirements	2.0	1
E7	Part-time staff	-1.0	0
E8	Difficult programming language	-1.0	4
			Total (EF):

< 3

< 3

< 3

> 3



**PF =
28 ph/UCP**

$$\begin{aligned} \text{Eff(UCP)} &= (13 + 100) * 1.02 * 1.085 * 28 \text{ ph} = \\ &= 3501 \text{ ph} \sim 22 \text{ pm} \end{aligned}$$

Discussion

How realistic is the approach?

What are its prerequisites/limitations?

...?

