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).

**UUCP = UAW + 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

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

\[
UAW = (\text{Total No. of Simple actors} \times 1) + (\text{Total No. Average actors} \times 2) + (\text{Total No. Complex actors} \times 3)
\]
## Unadjusted Use Case Weight

<table>
<thead>
<tr>
<th>Use Case Type</th>
<th>Description</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>1 to 3 transactions</td>
<td>5</td>
</tr>
<tr>
<td>Average</td>
<td>4 to 7 transactions</td>
<td>10</td>
</tr>
<tr>
<td>Complex</td>
<td>8 or more transactions</td>
<td>15</td>
</tr>
</tbody>
</table>

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

← 2. Tenant/Landlord closes the door
   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
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)

(adjusted) \[ UCP = UUCP \times TCF \times ECF \]

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

\[ UUCP = UAW + UUCW \]
Technical Complexity Factor – TCF

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

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

\[
TCF = 0.6 + \frac{TF}{100} \text{ with } TF = \sum_{i=1}^{13} (TF(i) \times Weight(i))
\]

LTAT.05.003 / Lecture 03/07 / © Dietmar Pfahl 2019
## Environmental Complexity Factor - ECF

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

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) with EF = \( \sum_{i=1}^{8} (EF(i) \times Weight(i)) \)
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)

\[ UCP = UUCP \times TCF \times ECF \]

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

\[ UUCP = UAW + UUCW \]
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):
  \[
  \text{Effort}(UCP) = UUCP \times TCF \times ECF \times PF
  \]

- UUCP is the sum of Unadjusted Actor Weights (UAW) and Unadjusted Use Case Weights (UUCW).
  \[
  UUCP = UAW + UUCW
  \]
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)
- Effort Estimate uses the Productivity Factor:
  \[ \text{Effort}(UCP) = \text{UUCP} \times \text{TCF} \times \text{ECF} \times \text{PF} \] [person-hours]
- UUCP is the sum of Unadjusted Actor Weights (UAW) and Unadjusted Use Case Weights (UUCW).
  \[ \text{UUCP} = \text{UAW} + \text{UUCW} \]

Either 20 person-hours/UCP or 28 person-hours/UCP
Productivity Factor - PF

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

\[ \text{PF} = 20 \text{ ph/UCP} \]

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

\[ \text{PF} = 28 \text{ ph/UCP} \]

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


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

EF values: 0, 1, 2, 3, 4, 5
Use Case Point Calculation and Effort Estimation Example
5 Steps to perform …

1. Determine Actor Weights
2. Determine Use Case Weights
3. Adjust for Technical Factors
4. Adjust for Environmental Factors
5. Calculate Productivity Factor
UCP-Example: Online Shop

Use Case Model

- 5 Actors
- 9 UCs

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

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

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

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

<table>
<thead>
<tr>
<th>Use Case Type</th>
<th>Description</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>1 to 3 transactions</td>
<td>5</td>
</tr>
<tr>
<td>Average</td>
<td>4 to 7 transactions</td>
<td>10</td>
</tr>
<tr>
<td>Complex</td>
<td>8 or more transactions</td>
<td>15</td>
</tr>
</tbody>
</table>

\[
UUCW = (\text{Total No. of Simple Use Cases} \times 5) + (\text{Total No. Average Use Case} \times 10) + (\text{Total No. Complex Use Cases} \times 15)
\]
2. Determine UC Weights (UUCW)

Use Case UC-2: Lock

<table>
<thead>
<tr>
<th>Related Requirements</th>
<th>REQ1, REQ2, and REQ5 stated in Table 2-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiating Actor</td>
<td>Any of: Tenant, Landlord, or Timer</td>
</tr>
<tr>
<td>Actor’s Goal</td>
<td>To lock the door &amp; get the lights shut automatically (?)</td>
</tr>
<tr>
<td>Participating Actors</td>
<td>LockDevice, LightSwitch, Timer</td>
</tr>
<tr>
<td>Preconditions</td>
<td>The system always displays the menu of available functions.</td>
</tr>
<tr>
<td>Postconditions</td>
<td>The door is closed and lock armed &amp; the auto-lock timer is reset.</td>
</tr>
</tbody>
</table>

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 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 to arm the lock, (d) signal to Timer to reset the auto-lock counter, and (e) signal to Timer to reset the alarm counter
← 3. System (a) signals affirmation to the Tenant/Landlord, (b) signals to LockDevice to arm the lock, (c) signals to Timer to reset the auto-lock counter, and (d) signals to Timer to reset the alarm counter
2. Determine UC Weights (UUCW)

<table>
<thead>
<tr>
<th>Use Case UC-2: Lock</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Related Requirements:</strong></td>
</tr>
<tr>
<td><strong>Initiating Actor:</strong></td>
</tr>
<tr>
<td><strong>Actor’s Goal:</strong></td>
</tr>
<tr>
<td><strong>Participating Actors:</strong></td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
</tr>
<tr>
<td><strong>Postconditions:</strong></td>
</tr>
</tbody>
</table>

**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 already armed), (c) signal to **Timer** to reset the auto-lock counter, and (d) signals to **LightSwitch** to turn the light off (?)

2. **Tenant/Landlord** closes the door
   - **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

**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** 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 \( \Rightarrow \) UC is average
2. Determine UC Weights (UUCW)

<table>
<thead>
<tr>
<th>Use Case UC-2:</th>
<th>Lock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related Requirements:</td>
<td>REQ1, REQ2, and REQ5 stated in Table 2-1</td>
</tr>
<tr>
<td>Initiating Actor:</td>
<td>Any of: Tenant, Landlord, or Timer</td>
</tr>
<tr>
<td>Actor’s Goal:</td>
<td>To lock the door &amp; get the lights shut automatically (?)</td>
</tr>
<tr>
<td>Participating Actors:</td>
<td>LockDevice, LightSwitch, Timer</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>The system always displays the menu of available functions.</td>
</tr>
<tr>
<td>Postconditions:</td>
<td>The door is closed and lock armed &amp; the auto-lock timer is reset.</td>
</tr>
<tr>
<td>Flow of Events for Main Success Scenario:</td>
<td></td>
</tr>
<tr>
<td>→</td>
<td>1. Tenant/Landlord selects the menu item “Lock”</td>
</tr>
<tr>
<td>→</td>
<td>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 (?)</td>
</tr>
<tr>
<td>←</td>
<td>2. Tenant/Landlord closes the door</td>
</tr>
<tr>
<td>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</td>
<td></td>
</tr>
</tbody>
</table>

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

UUCW = 2x5 + 3x10 + 4x15 = 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:

\[
TCF = 0.6 + \frac{TF}{100} \text{ with } TF = \sum_{i=1}^{13} (TF(i) \times Weight(i))
\]
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)

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

**TCF = 0.6 + \frac{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:

\[
ECF = 1.4 + (-0.03*EF) \quad \text{with} \quad EF = \sum_{i=1}^{8} (EF(i) \times Weight(i))
\]
4. Adjust for Environmental Factors (ECF)

Let’s assume:

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

• $\text{ECF} = ?$
4. Adjust for Environmental Factors (ECF)

<table>
<thead>
<tr>
<th>Factor</th>
<th>Description</th>
<th>Weight</th>
<th>Assigned Value</th>
<th>Weight x Assigned Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>Familiarity with development process used</td>
<td>1.5</td>
<td>3</td>
<td>4.5</td>
</tr>
<tr>
<td>E2</td>
<td>Application experience</td>
<td>0.5</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>E3</td>
<td>Object-oriented experience of team</td>
<td>1.0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>E4</td>
<td>Lead analyst capability</td>
<td>0.5</td>
<td>5</td>
<td>2.5</td>
</tr>
<tr>
<td>E5</td>
<td>Motivation of the team</td>
<td>1.0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>E6</td>
<td>Stability of requirements</td>
<td>2.0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>E7</td>
<td>Part-time staff</td>
<td>-1.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>E8</td>
<td>Difficult programming language</td>
<td>-1.0</td>
<td>4</td>
<td>-4</td>
</tr>
</tbody>
</table>

**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)

<table>
<thead>
<tr>
<th>Factor</th>
<th>Description</th>
<th>Weight</th>
<th>Assigned Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>Familiarity with development process used</td>
<td>1.5</td>
<td>3</td>
</tr>
<tr>
<td>E2</td>
<td>Application experience</td>
<td>0.5</td>
<td>3</td>
</tr>
<tr>
<td>E3</td>
<td>Object-oriented experience of team</td>
<td>1.0</td>
<td>2</td>
</tr>
<tr>
<td>E4</td>
<td>Lead analyst capability</td>
<td>0.5</td>
<td>5</td>
</tr>
<tr>
<td>E5</td>
<td>Motivation of the team</td>
<td>1.0</td>
<td>2</td>
</tr>
<tr>
<td>E6</td>
<td>Stability of requirements</td>
<td>2.0</td>
<td>1</td>
</tr>
<tr>
<td>E7</td>
<td>Part-time staff</td>
<td>-1.0</td>
<td>0</td>
</tr>
<tr>
<td>E8</td>
<td>Difficult programming language</td>
<td>-1.0</td>
<td>4</td>
</tr>
</tbody>
</table>

Total (EF):

\[
\text{Eff(UCP)} = (13 + 100) \times 1.02 \times 1.085 \times 28 \text{ ph} = 3501 \text{ ph} \approx 22 \text{ pm}
\]
Discussion

How realistic is the approach?
What are its prerequisites/limitations?
...