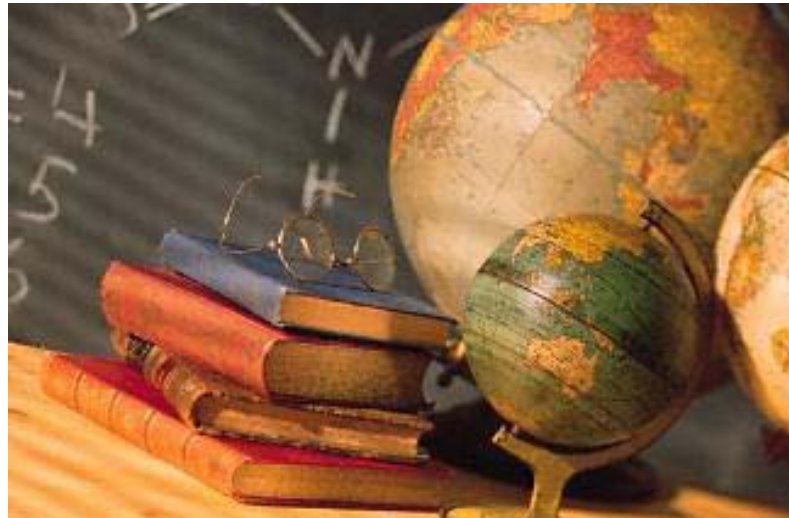




**ÉCOLE DE TECHNOLOGIE SUPÉRIEURE – MONTRÉAL - CANADA**



# **An Evaluation of the Design of Use Case Points (UCP)**



**Joost Ouwerkerk & Alain Abran**

©Ouwerkerk-Abran 2006 MENSURA 2006, Cadiz (Spain), Nov. 6-8 1



## **List of topics**

- 1) Overview of Use Case Points**
- 2) Published empirical evidence**
- 3) Analysis of the Design of Use Case Points**
- 4) Discussion**



## Origin:

- ➔ **Use Case: Jacobson – 1987**
- ➔ **Integration into RUP – 1995**
- ➔ **Use Case Points – Kraner 1993**



| <b>Table 1: ACTOR Weights</b> |   |               |
|-------------------------------|---|---------------|
| <b>Complexity</b>             | <b>Definition.</b>  | <b>Weight</b> |
| <b>Simple</b>                 | System interaction via API.   | <b>1</b>      |
| <b>Average</b>                | Average interaction system via protocol, or Human interaction via a command line. | <b>2</b>      |
| <b>Complex</b>                | Complex human interaction via a graphical user interface                          | <b>3</b>      |

| <b>Table 2: USE CASE Weights</b> |   |               |
|----------------------------------|---|---------------|
| <b>Complexity</b>                | <b>Definition</b>                                       | <b>Weight</b> |
| <b>Simple</b>                    | 3 transactions or fewer;<br>5 analysis classes or fewer | <b>5</b>      |
| <b>Average</b>                   | 4 to 7 transactions;<br>5 to 10 analysis classes        | <b>10</b>     |
| <b>Complex</b>                   | Over 7 transactions;<br>Over 10 analysis classes        | <b>15</b>     |



**Table 3: Technical Quality Factors – TCF**

| <b>Factor</b> | <b>Description</b>                  | <b>Weight</b> |
|---------------|-------------------------------------|---------------|
| <b>F1</b>     | Distributed system                  | <b>2</b>      |
| <b>F2</b>     | Performance (response time or flow) | <b>1</b>      |
| <b>F3</b>     | Efficiency of user interface        | <b>1</b>      |
| <b>F4</b>     | Processing complexity               | <b>1</b>      |
| <b>F5</b>     | Reusability                         | <b>1</b>      |
| <b>F6</b>     | Installability                      | <b>0.5</b>    |
| <b>F7</b>     | Operability                         | <b>0.5</b>    |
| <b>F8</b>     | Portability                         | <b>2</b>      |
| <b>F9</b>     | Maintenability                      | <b>1</b>      |
| <b>F10</b>    | Simultaneous access                 | <b>1</b>      |
| <b>F11</b>    | Security                            | <b>1</b>      |
| <b>F12</b>    | Direct access for third parties     | <b>1</b>      |
| <b>F13</b>    | Training features or online help    | <b>1</b>      |

**Table 4: Environmental Factors – EF**

| <b>Factor</b> | <b>Description.</b>                         | <b>Weight</b> |
|---------------|---|---------------|
| <b>F1</b>     | Familiarity with the methodology            | <b>1.5</b>    |
| <b>F2</b>     | Part-time status                            | <b>-1</b>     |
| <b>F3</b>     | Analysis capability                         | <b>0.5</b>    |
| <b>F4</b>     | Experience with the application             | <b>0.5</b>    |
| <b>F5</b>     | Experience with object-oriented methodology | <b>1</b>      |
| <b>F6</b>     | Motivation                                  | <b>1</b>      |
| <b>F7</b>     | Difficulty of the programming language      | <b>-1</b>     |
| <b>F8</b>     | Stability of the specifications             | <b>2</b>      |



Formula to calculate the number of UCP:

☞ **UCP = UUCP \* TCF \* EF**

- **UUCP = Unadjusted Use Case Points**
- **TCF = Technical Quality Factor**
- **EF = Environmental Factor**





## List of topics

1) Overview of Use Case Points

**2) Published empirical evidence**

3) Analysis of the Design of Use Case  
Points

4) Discussion



## Related Studies

- Nageswaren – 2001:
  - Adaptation
  - 1 single project
- Mohagheghi – 2005
  - Adaptation
  - 2 projects
- Carroll – 2005
  - Addition of a risk coefficient
  - Claim of 200 projects over 5 years, but no documented evidence & no detailed analysis





## List of topics

- 1) Overview of Use Case Points
- 2) Published empirical evidence
- 3) Analysis of the Design of Use Case Points
- 4) Discussion



**Table 5: ENTITIES**

| <b>Entity</b>                 | <b>Description</b>   |
|-------------------------------|--|
| Actor                         | A use case, as defined by [JAC87], describes the interaction between the actors and the system. The actor is any agent (machine or human) that acts upon system functionality. |
| Use Case                      | A simple functional requirement description for a specific goal, written in the form of a sequence of interactions between an actor and the system.                            |
| Specification of requirements | The set of planned requirements for a system, including the functional requirements (written in use-case form) and other non-functional requirements.                          |
| Development team              | The human resources participating in the project of designing, programming and testing the system.   |
| Programming language          | The computer programming language used by the development team to code the software system (Java or C++, for instance).  |

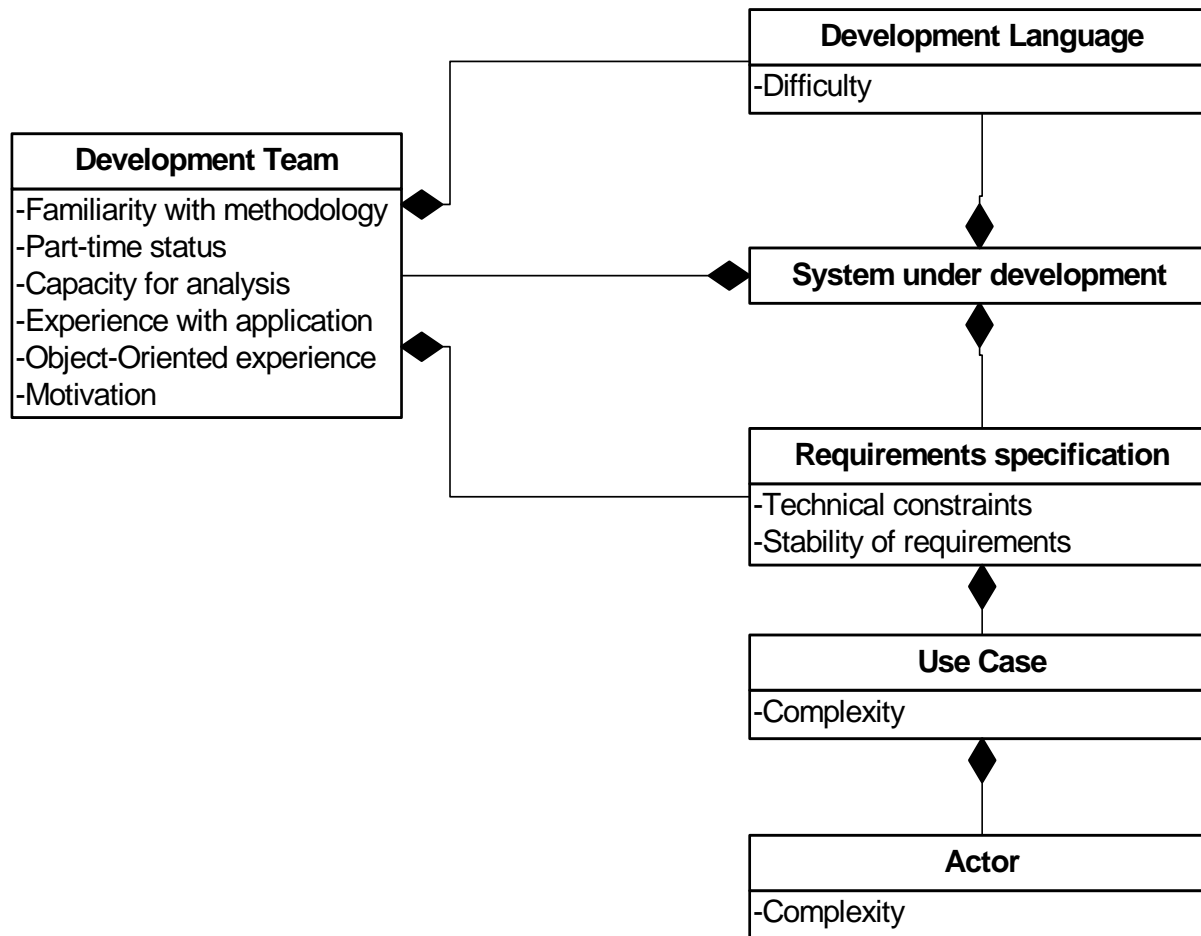


Figure 1: Set of entities and attributes measured in UCP



**Table 6: ATTRIBUTES**

| <b>Entity</b>                 | <b>Attribute</b>                                | <b>Measurement rule</b>  |
|-------------------------------|---|--|
| Actor                         | Complexity (of actor)                           | The type of complexity (simple, average or complex) of the interaction between the actor and the system. |
| Use case                      | Complexity (of use case)                        | The type of complexity (simple, average or complex) measured in the number of transactions.              |
| Specification of requirements | Relevance of the technical quality requirements | The level of relevance (from 0 to 5) of each of the 13 known non-functional qualities                    |
|                               | Stability of requirements                       | The level of stability (from 0 to 5) of the functional and non-functional requirements                   |
| Development team              | Familiarity with the methodology                | The level (from 0 to 5) of skills and knowledge of the development methodology in use for the project.   |
|                               | Part-time status                                | The level (from 0 to 5) of part-time staff on the team   |
|                               | Analysis capability                             | The level (from 0 to 5) of analysis capabilities of the development team with respect to project needs.  |
|                               | Application experience                          | The level (from 0 to 5) of team experience with the application domain of the system                     |
|                               | Object-oriented experience                      | The level (from 0 to 5) of team experience with object-oriented design                                   |
|                               | Motivation.                                     | The level (from 0 to 5) of team motivation   |
| Programming language          | Difficulty                                      | The level (from 0 to 5) of programming difficulty  |



## **Design issues:**

- ☞ Number of 'things' being measured = 7**
- ☞ Number of properties being measured = 11**
- ☞ Based on Use Cases = lack of consistency**



“Complexity” attribute, assigned to actors and use cases:

- Categorized as being of the ordinal scale type using a scale of three values: simple, average and complex.
  - Thus an actor categorized by the measurer as “simple” is considered less complex than an “average” actor, and an “average” actor less complex than a “complex” actor.
- The scale is similar for use cases: the same category labels are used (simple, average and complex)
  - however it cannot be assumed that the categories and the categorization process are similar, since different entity types are involved.





Technical and resource factors are also all evaluated through a categorization process on an ordinal scale, but one with integers from 0 to 5 (0, 1, 2, 3, 4 or 5)

- These numbers do not represent numerical values on a ratio scale, but merely a category on an ordinal scale; that is, they are merely ordered labels and not numbers.
  - ☞ Thus, a programming language assigned a difficulty level of 1 is considered to be less difficult for the development team than a programming language of difficulty level 2, but cannot be considered to be exactly one unit less difficult than one categorized as having a difficulty level of 2 because these levels are being measured on an ordinal scale.
- No justification provided in the description of the UCP model to support a ratio scale:
  - ☞ for example, that for a programming language of factor 4 is it twice as difficult is a programming language of factor 2?
  - ☞ The levels must therefore be regarded as being on an interval scale.
  - ☞ It must also be noted that, even though they have the same labels, e.g. 1,2,3, etc., the intervals are not necessarily regular; for example, each label might represent a different interval, and each interval may not be, and does not need to be, regular within an attribute being measured.



## ***Use-case complexity:***

- ☞ *The UCP model transforms the measurements of use-case complexity:*
  - *from a ratio-type scale (the number of transactions or classes of analysis)*
  - *into an ordinal-type scale (complexity categories),*
  - *and then back to a ratio-type scale (UCP weights).*
    - ☞ *The arbitrary assignment of the weights (5 for simple, 10 for average and 15 for complex) could have been avoided if the number of transactions or classes of analysis had been kept as numbers on a ratio-type scale rather than losing this quantitative precision by mapping them to only three ordered categories with arbitrarily assigned values of 5, 10 and 15.*



- ☞ **+ Inadmissible transformations across scale types:**
  - Nominal
  - Ordinal
  - Ratios



## **End results of Use Case Points:**

- ☞ Actors & Use Cases & Specs & Dev. Team & Programming language = ?**
  
- ☞ Unknown measurement unit:**
  - It cannot be a number of Use Cases and**
  - It cannot be a number of days**



## List of topics

- 1) Overview of Use Case Points
- 2) Published empirical evidence
- 3) Design of Use Case Points
- 4) Discussion



## Discussion

- ☞ What would you recommend to practitioners?
- ☞ What would you recommend to your students?





**ÉCOLE DE TECHNOLOGIE SUPÉRIEURE – MONTRÉAL - CANADA**



***Thank You !***



**alain.abran@etsmtl.ca**

**joosto@gmail.com**