DA-IICT

IT314: Software Engineering

Software Effort Estimation

Use Case Points

Saurabh Tiwari

1

Estimation?

- A large proportion of industrial systems development projects significantly
  - overrun budget or
  - are delivered after schedule (or not delivered at all), or
  - are not delivered with the specified functionality.

→ Need for early and precise effort estimates.
→ Can use case models be used to improve estimation?

Approach...

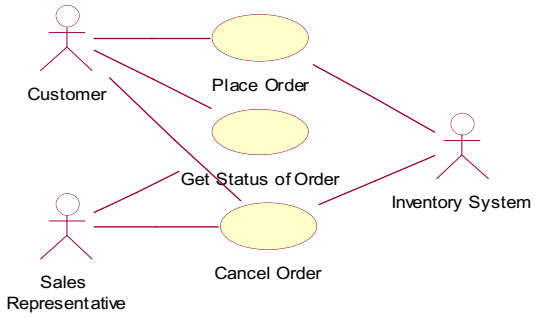
- A use case model defines the functional scope of the system to be developed. The functional scope is the basis for top-down estimation.
- Estimation parameters can be derived from a use case model.
- Following a use case driven development process, a high-level use case model is available in the inception phase, and a detailed use case model is available at the start of the elaboration phase.
- Many companies use a system's use case model in the estimation process.
- How can a use case model best be applied in estimating software development effort ?

Use Case Modeling

A use case model describes a system's intended functions and its environment. It has two parts:

1. A diagram that provides an overview of actors and use cases, and their interactions.
   - An actor represents a role that the user can play with regard to the system.
   - A use case represents an interaction between an actor and the system.
2. The use case descriptions detail the requirements by documenting the flow of events between the actors and the system.

Example of Use Case Diagram



Example of Use Case Description

Use Case Name: Place Order

Short description:
The customer provides address information and a list of product codes.
The system confirms the order.

Basic flow of events:

1. Customer enters name and address
2. Customer enters product codes for items he wishes to order
3. The system will supply a product description and price for each item
4. The system will keep a running total of items ordered as they are entered
5. The customer enters credit card information
6. The system validates the credit card information
7. The system issues a receipt to the customer

Alternative flow of events:

3.1 The product is out of stock:
- 3.1.1 The systems informs the customer that the product can not be ordered.

6.1 The credit card is invalid
- 6.1.1 The system informs the customer that his credit card is invalid
- 6.1.2 The customer can enter credit card information again or cancel the order.

Pre-Conditions: The customer is logged on to the system

Post-Conditions: The order has been submitted

Extension Points: None

Use Case Point (UCP)

- The UCP considers the Use Case (UC) model to estimate the size and effort of an object-oriented system.
- The UCP is an adaptation from FP and MKII FP (United Kingdom Software Metrics Association (UKSMA)).

Estimation Method

- Each actor and use case is categorized according to complexity and assigned a weight.
  - The complexity of a use case is measured in number of transactions.
- The unadjusted use case points are calculated by adding the weights for each actor and use case.
- The unadjusted use case points are adjusted based on the values of 13 technical factors and 8 environmental factors.
- Finally the adjusted use case points are multiplied with a productivity factor.

Estimation Method

Use Case Classification	No. of Transactions	Weight
Simple	1 to 3 transactions	5
Average	4 to 7 transactions	10
Complex	8 or more transactions	15

UUCW = (Total No. of Simple Use Cases x 5) + (Total No. Average Use Case x 10) + (Total No. Complex Use Cases x 15)

Actor Classification	Type of Actor	Weight
Simple	External system that must interact with the system using a well-defined API	1
Average	External system that must interact with the system using standard communication protocols (e.g. TCP/IP, FTP, HTTP, database)	2
Complex	Human actor using a GUI application interface	3

UAW = (Total No. of Simple actors x 1) + (Total No. Average actors x 2) + (Total No. Complex actors x 3)

Adjust Based on Technical Factors

Factor number	Factor description	Weight
T1	Distributed system	2
T2	Response or throughput performance objective	1
T3	End-user efficiency	1
T4	Complex internal processing	1
T5	Code must be reusable	1
T6	Easy to install	0.5
T7	Easy to use	0.5
T8	Portable	2
T9	Easy to change	1
T10	Concurrent	1
T11	Includes special security features	1
T12	Provides direct access for third parties	1
T13	Special user training facilities are required	1

Adjust Based on Environmental Factors

Factor number	Factor description	Weight
F1	Familiar with RUP	1.5
F2	Application experience	0.5
F3	Object-oriented experience	1
F4	Lead analyst capability	0.5
F5	Motivation	1
F6	Stable requirements	2
F7	Part-time workers	-1
F8	Difficult programming language	-1

Producing an Estimate

- The unadjusted actor weight, UAW, is calculated adding the weights for each actor.
- The unadjusted use case weights, UUCW, is calculated correspondingly.
- The unadjusted use case points, UUCP, = UAW + UUCW.
- The technical factor, TCF, = .6 + (.01*$\Sigma_{1..13}T_n$*$Weight_n$).
- The environmental factor, EF, = 1.4 + (-.03* $\Sigma_{1..8}F_n$*$Weight_n$).
- UCP = UUCP*TCF*EF
- Estimate = UCP * Productivity factor

Results from Case Studies

Company	Project	Use Case Estimate	Expert Estimate	Actual Effort	Deviation use case est.	Deviation exp. est.
Mogul	A	2550	2730	3670	-31%	-26%
Mogul	B	2730	2340	2860	-5%	-18%
Mogul	C	2080	2100	2740	-24%	-23%
CGE&Y	A	10831	7000	10043	+8%	-30%
CGE&Y	B	14965	12600	12000	+25%	+5%
IBM	A	4086	2772*	3360	+22%	-18%
Students project	A	666		742	-10%	
Students project	B	487		396	+23%	
Students project	C	488		673	-25%	

Characteristics of Projects

Project	Estimate produced	Use case model	Characteristics
Mogul – A	Before	Detailed	Duration = 7 months, Team = 6, Development tools = Java and Web-logic Application domain = Banking
Mogul – B	After	Detailed	Duration = 3 months, Team = 6, Development tools = MSVisual studio Application domain = CRM within banking
Mogul – C	After	Sequence diagrams	Duration = 4 months, Team = 5, Development tools = Java and Web-logic Application domain = Banking
CGE&Y – A	After	No details	Duration = 7 months, Team = 6, Development tools = Java2Enterprise and Websphere, Application domain = Internet application for banking
CGE&Y – B	Before	Detailed	Duration = 9 months, Team = 3 - 4, Development tools = C++, Application domain = Real-time system, part of larger commercial system

Characteristics cont.

Project	Estimate produced	Use case model	Characteristics
IBM	After	Detailed	Duration = 3 months, Team = 8, Development tools = Smalltalk, Java and C++, Application domain = Internet solution for home shopping.
Students project	Before	Detailed	Duration = 2 months, Team = 6, Development tools = Java and C++, Application domain = News service on the internet.
Students project	Before	Detailed	Duration = 2 months, Team = 5, Development tools = Java, Application domain = Travelling information service for wap.
Students project	Before	Detailed	Duration = 2 months, Team = 5, Development tools = Java and IDEA 2.0, Application domain = Generating source code from UML .

Prerequisites for Applying the UCP Method

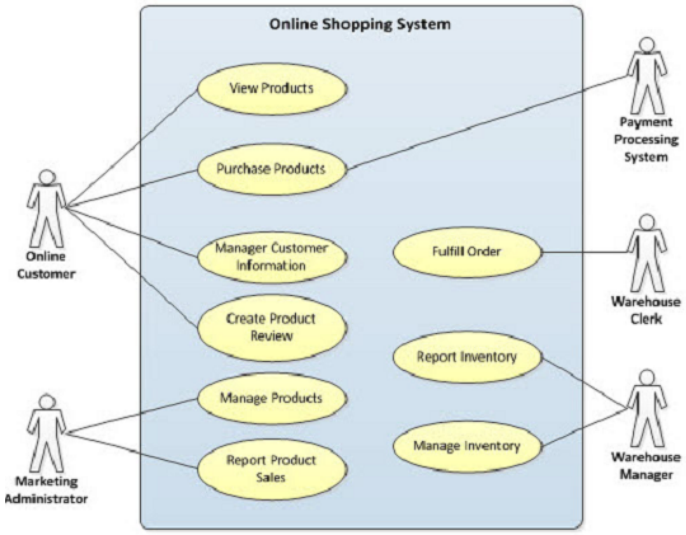
1. Correctness of the use case model:

   The use case model should include the functional requirements of all the user groups. The main challenge is sufficient access to skilled and motivated domain experts.

2. Level of detail:

   The use case model should be described at an appropriate level of detail. The main challenges are to obtain balanced use cases and avoid ”infinite” expansion. Possible solutions are guidelines and good examples of use case models.

Example !!



UUCW & UAW

UUCW = (Total No. of Simple Use Cases x 5) + (Total No. Average Use Cases x 10) + (Total No. Complex Use Cases x 15)

For the Online Shopping System, the UUCW = (2 x 5) + (3 x 10) + (4 x 15) = 100

UAW = (Total No. of Simple Actors x 1) + (Total No. Average Actors x 2) + (Total No. Complex Actors x 3)

For the Online Shopping System, UAW = (1 x 1) + (0 x 2) + (4 x 3) = 13

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
			Total (TF):	**42**

TCF = 0.6 + (TF/100)

For the Online Shopping System, TCF = 0.6 + (42/100) = 1.02

TCF = 1.02

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**

Next, the ECF is calculated:

ECF = 1.4 + (-0.03 x EF)

For the Online Shopping System, ECF = 1.4 + (-0.03 * 10.5) = 1.085

ECF = 1.085

UCP = (UUCW + UAW) x TCF x ECF

For the Online Shopping System, UCP = (100 + 13) x 1.02 x 1.085 = 125.06

UCP = 125.06

Now that the size of the project is known, the total effort for the project can be estimated. For the Online Shopping System example,
28 man hours per use case point will be used.

Estimated Effort = UCP x Hours/UCP

For the Online Shopping System, Estimated Effort = 125.06 x 28

Estimated Effort = 3501 Hours

Limitations of UCP

UCs in UCP could only be classified into one of the three categories
•Simple
•Average
•Complex
(no matter how many sections it has or how big it is)

For example, consider three UCs: UC1, UC2 and UC3 with respectively 70, 2 and 8 transactions.

According to the UCP classification tables,
•UC2 is considered simple
•UC1 and UC3 are complex.

However, UC1 is much more complex than UC3.

Use Case Size Points (USP)

- Applied to each UC separately
- It is possible to estimate time and cost for the development of a particular UC and not only for the whole system
- USP measures the functionality by considering the structures and sections of a UC
  - counting the number and weight of scenarios, actors, precondition and postconditions.
- USP components
  - Actors
  - Pre-conditions
  - Post-conditions
  - Flow of events (Basic, Alternative)
  - Exception

Actors Classification

- Each actor has its complexity (CA) determined according to the data provided to or received from the UC being classified.
- The total complexity of actors in the UC (TPA) is calculated by

$$TPA = \sum_{i=1}^{n} CA_i$$

n - Number of actors in the UC

Complexity	Data	UUSP
Simple	≤ 5	2
Average	6 to 10	4
Complex	> 10	6

Precondition Classification

- Each precondition of the UC has the complexity (CPrC) determined according to the number of logical expressions tested by the condition
- The total complexity of the preconditions (TPPrC) is given by

$$TPPrC = \sum_{i=1}^{n} CPrC_i$$

n - Number of preconditions in the UC

Complexity	Tested Expression	UUSP
Simple	1 logical expression	1
Average	2 or 3 logical expressions	2
Complex	3 logical expressions	3

Postcondition Classification

- The complexity of the postconditions (CPoC) is determined according to the number of related entities
- The total com-plexity of postconditions (TPPoC) is given by

$$TPPoC = \sum_{i=1}^{n} CPoC_i$$

n - Number of postconditions in the UC

Complexity	Entities	UUSP
Simple	≤ 3	1
Average	4 to 6	2
Complex	> 6	3

Main Scenario Classification

- The main scenario must be classified according to its number of entities and to the number of elementary steps needed to the scenario conclusion.
- The complexity of the scenario (PCP) is given by the sum of both values (number of entities + number of steps)

Complexity	Entities + Steps	UUSP
Very Simple	≤ 5	4
Simple	6 to 10	6
Average	11 to 15	8
Complex	16 to 20	12
Very Complex	> 20	16

Alternative Scenario Classification

- All the alternative scenarios are classified similarly to the main scenario.
- Each alternative scenario receives a number of points (PCA) according to

Complexity	Entities + Steps	UUSP
Very Simple	≤ 5	4
Simple	6 to 10	6
Average	11 to 15	8
Complex	16 to 20	12
Very Complex	> 20	16

- The total complexity of the alternative scenarios (TPCA) is given by

$$TPCA = \sum_{i=1}^{n} PCA_i$$

n - Number of alternative scenario in the UC

Exception Classification

- Each exception present in the UC must also be analyzed according to its complexity (CE), determined by the number of logical expressions tested to detect the exception occurrence.
- The total points added by exceptions (TPE) are determined by

$$TPE = \sum_{i=1}^{n} CE_i$$

n - Number exceptions in the UC

Complexity	Tested Expressions	UUSP
Simple	1 logical expression	1
Average	2 or 3 logical expressions	2
Complex	$>$ 3 logical expressions	3

Unadjusted Use-case Size Point (UUSP)

- The Unadjusted Use-case Size Point (UUSP) is given by the sum of the complexity values of all sections of the UC

$$UUSP = TPA + TPPrC + PCP + TPCA + TPE + TPPoc$$

Application of the adjustment factor (value between 0 to 5)

Factor	Requirement	Influence
F1	Data communication	I1
F2	Distributed processing	I2
F3	Performance	I3
F4	Equipment utilization	I4
F5	Transaction Capacity	I5
F6	On-line input of data	I6
F7	User efficiency	I7
F8	On-line update	I8
F9	Code reuse	I9
F10	Complex processing	I10
F11	Easiness of deploy	I11
F12	Easiness operation	I12
F13	Many places	I13
F14	Facility of change	I14

Technical Factors
(adapted from FP)

$$FTA = 0.65 + (0.01 * \sum_{i=1}^{14} I_i)$$

Factor	Description	Influence
E1	Formal development process existence	I1
E2	Experience with the application being developed	I2
E3	Experience of the team with the used technologies	I3
E4	Presence on an experienced analyst	I4
E5	Stable requirements	I5

Environmental Factors

$$FAA = (0.01 * \sum_{i=1}^{5} I_i)$$

Adjusted USP

$$USP = UUSP * (FTA - FAA)$$

Next lecture….
PERT Chart
Project Planning Document
Testing Plans
Testing Strategy

Example:
Source- https://en.wikipedia.org/wiki/Use_Case_Points