



# TRUE PROGRAM SUCCESS '05

PRICE Systems' US Symposium  
Grand Hyatt Tampa Bay  
Tampa Bay, Florida  
March 23-25, 2005

## Estimating from Use Cases

Arlene Minkiewicz  
Chief Scientist  
PRICE Systems, LLC



## Conclusion

- > Use Cases have been an effective language for describing requirements of a software system in a way that facilitates communication between developers of software systems and the end users
- > Use Cases are available before significant architectural and implementation decisions have been made
- > Use Cases can be an effective measure for estimating software size early on in a software project before enough information is available to estimate size using more traditional measures
- > Estimates based on use cases alone are not as accurate as those possible after architectural and implementation decisions have been made



# Use Case Sizing

1. The Problem – Historical Perspective
2. Solution Methodology
3. Use Cases – What are they and what are they good for?
4. Use Case Conversion Points
  - Development
  - Uses
5. Challenges, Limitations and Risks
6. Conclusions

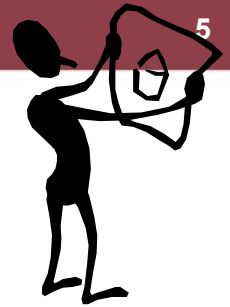


## The Problem – an historical perspective

- > Early cost estimators measured software size in Source Lines of Code (SLOC)
  - Made sense when projects used similar programming languages and functional characteristics
  - Third generation languages and functional programming practices
  - Easy to count and verify with finished software systems
  
- > Improved software development practices, processes and tools have lead to other more desirable metrics
  - Function Points
  - Object Points



Today – not only to project managers want metrics more in line with practices and tools, they want to be able to perform estimates earlier in the software lifecycle, before implementation and architectural decisions have been made



## Solution Methodology

- > Understand and bound the problem
- > Understand what information is available at the Use Case stage of a project
- > Identify a verifiable relationship between use case information and an accepted size metric – Function Points
- > Construct a mathematical model to predict Function Points size from Used Case Information
- > Test the mathematical model against actual data and refine the model

# Use Cases – What Are They and What Are They Good For?

- > First Introduced by Ivar Jacobson in the mid 1980's
- > Provide a language for software people to effectively communicate with software users
  - Software Requirements translated to English
- > Each Use Case Describes a typical interaction that may occur between user and software
  - User may be a human user or another software system
  - Focus is on functions users want to perform not on how the system will perform them

# Components of a Use Case

- > Each Use Case relates to a specific and complete instance of behavior or transaction of the system
  
- > Each Use Case is described by
  - Use Case Name
    - Describes the goal of the use case
    - Usually in the form of an 'action verb + object'
  - Actors involved in the specific behavior described
    - Actors represent the roles that depend on the transaction or that the transaction depends on
    - Either a person or another software system
  - Steps in the use case description
    - List of the things that the software must do to complete the goal of the use case
    - At a top level these focus on successful satisfaction of the goal
    - More detailed use cases should follow to account for cases when circumstances prevent success

## Research Details

- > Detailed use case information available for approximately 20 components
- > Function Points counts performed by certified function point specialist
- > Data collected from top level use cases
  - Number of actors
  - Complexity of actors
  - Number of steps in the transaction description

## Research Details

- > Started with Use Case Points Developed by Gustav Karner
- > For each Use Case Determined
  - Complexity of the Use Case as a function of the number of transactions
    - Simple -  $\leq 3$  transactions
    - Average - between 4 and 7 transactions
    - High - between 8 and 13 transactions
    - Extra high - 14 or more transactions
- > Number of actors per complexity category
  - Simple actors – External system interfaced through API
  - Average actors – External system interfaces using text based protocol
  - Complex actors – End users interfacing through GUI

## Research Details

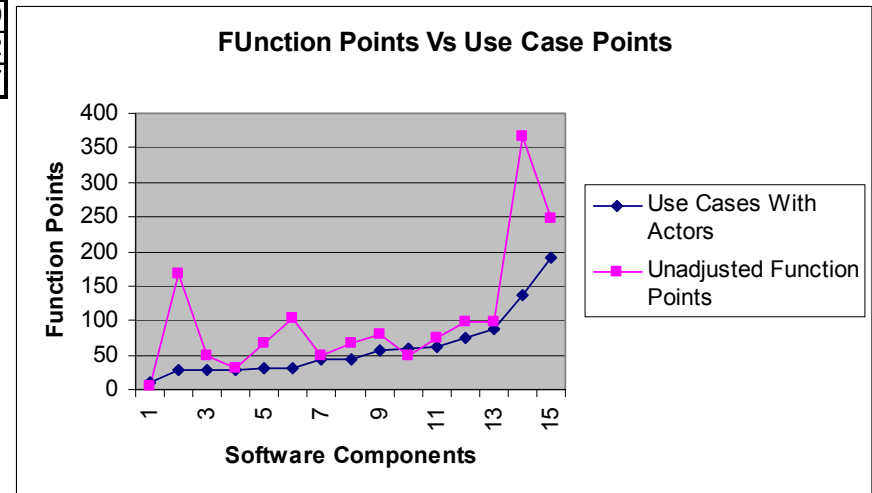
- > Assigned weights to Use Case and Actor complexity as follows
  
- > Use Cases
  - Simple = 5
  - Average = 10
  - High = 15
  - Very High = 20
  
- > Actors
  - Simple = 1
  - Average = 2
  - Complex = 3

## Research Details

- > Calculated Use Conversion Points using several different relationships
- > UCP with actors
  - = Weighted Use Case + Sum(Weighted Actors)
- > UCP with no actors
  - = Weighted Use Case
- > UCP with simple Actor
  - Weighted Use Case + Weighted (Actor Average)
- > Summed the UCP for all use cases associated with a Function Point Count

# Research Details – sample result

Name	UCP1	UCP2	UCP3	UFP	FP
Use Case 1	10	5	10	5	6
Use Case 2	28	25	28	168	188
Use Case 3	29	20	29	49	54
Use Case 4	29	20	23	31	34
Use Case 5	31	20	29	68	76
Use Case 6	31	25	30	102	115
Use Case 7	43	25	34	49	55
Use Case 8	45	30	36	68	76
Use Case 9	56	30	38	79	89
Use Case 10	59	35	44	48	53
Use Case 11	63	40	51	76	85
Use Case 12	76	70	76	98	109
Use Case 13	88	70	86	98	109
Use Case 14	136	115	120	366	412
Use Case 15	191	115	144	247	277



# Research Results – what questions to ask

afm test 6 (Tables and Generators)

PRICE True S Catalog

## USE CASE POINTS SIZE GENERATOR

Describe Use Cases

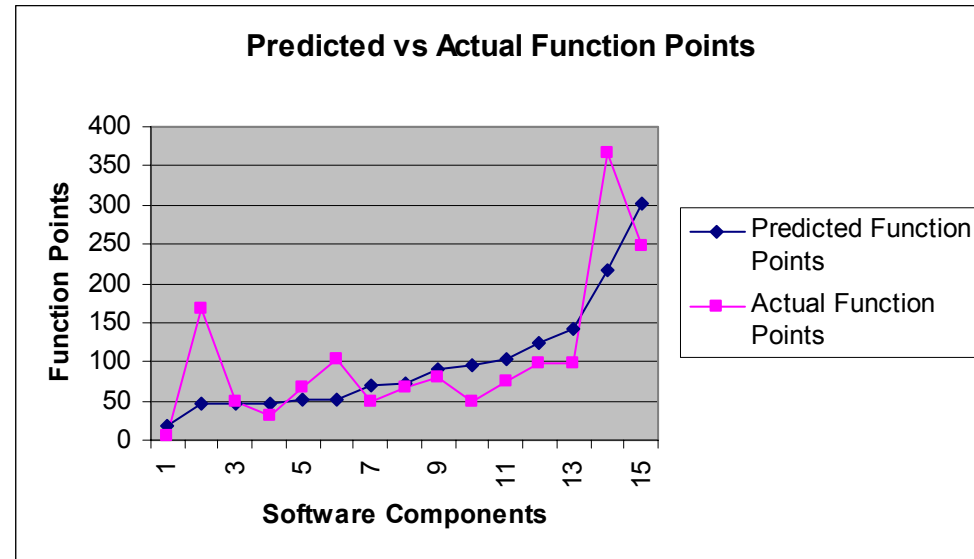
Use Case Name	Number of Transactions	Actors		
		Simple	Average	Complex
Use Case 1	0	0	0	0
Use Case 2	0	0	0	0
Use Case 3	0	0	0	0

Add Row      Delete Row

OK      Cancel

# Research Results

- > Relationship Determined for Use Case Points with Actors and Unadjusted Function Points – R-square = 0.6
- > Not a slam dunk but relationship shows promise
- > Currently studying outliers to determine missing factors in calculation
- > Looking for additional data to extend study



## Research Results - Limitations

- > Not always a one to one correspondence between use cases and function points
  - Counts are done at a high level
  - Overlaps need to be accounted for
- > Lacking a standard for use case definition there is subjectivity when determining the number of transactions
  - Need to “re-engineer” some use cases to accomplish consistent level of detail
- > Conversion to Function Points rather than direct relationship to effort adds an additional level of abstraction to an already abstracted process

## Conclusion

- > Use Cases have been an effect language for describing requirements of a software system in a way that facilitates communication between developers of software systems and the end users
- > Use Cases are available before significant architectural and implementation decisions have been made
- > Use Cases can be an effective measure for estimating software size early on in a software project before enough information is available to estimate size using more traditional measures
- > Estimates based on use cases alone are not as accurate as those possible after architectural and implementation decisions have been made

**Arlene F. Minkiewicz**  
**Chief Scientist**  
**PRICE Systems, L.L.C**  
**856-608-7222**  
**[Arlene.minkiewicz@pricesystems.com](mailto:Arlene.minkiewicz@pricesystems.com)**

Our mission is to  
provide true cost  
forecasting and  
modeling solutions  
that empower our  
clients to sustain  
profitable growth  
through faster  
better decisions



**PRICE**

FASTER DECISIONS. BETTER DECISIONS.™

## About PRICE Systems

- > **Leader in Program Affordability Management solutions**
- > **Combine cost estimating, project control, and knowledge management – *ensuring project success at every decision gateway***
- > **Customers increase visibility, minimize risk and cost, accelerate project development, and improve the effectiveness of project selection, control and delivery**

[www.PRICESystems.com](http://www.PRICESystems.com)



POWERED BY  
**PRICE** Research