

Use Stories vs UML Use Cases in Modular Transformation

Ravi Madanayake¹, G K A Dias², N D Kodikara³

^{1,2,3} University of Colombo School of Computing, Colombo, Sri Lanka

Abstract

There have been a considerable amount of compatibility issues when considering models in different system development methodologies. Among them are the low compatibility between the conceptual models, logical models and system models. A Survey done by us shows that greater percentage of companies use both UML as well as Non-UML techniques for their projects, (even in Agile processes) to help design their software. This means that there can be incompatibilities between them because their creators have different mindsets when constructing those models. Redundancy and unnecessary waste of time could be avoided, if other types of diagrams could be derived from one basic type. According to the survey done by us, majority seems to be using Use Case diagrams, Class diagrams, Entity Relationship Diagrams (ERD) and User Stories. Inter-compatibilities between User Stories in the Agile Methodologies and Use Case Models in the Unified Modelling Language (UML) are looked at in this paper.

Keywords: UML, User Stories, Use Cases, Modular Transformation

1.0 Introduction

When considering the models used in different systems development methodologies there are considerable amount of compatibility issues [1][2][3][4]. This means that there can be incompatibilities between them because their creators have different mindsets when constructing those models. This usually leads to a lot of confusion during the process of system development.

The objective of this publication is to compare User Stories with Use Case Models, for the purpose of modular transformation.

After a survey on the internet, we found out that the User Stories and Use Case models were from two different domains and were therefore totally non-compatible with each other. According to Mat Chacon [5], User stories are more efficient than use cases in agile processes such as Scrum and Use Cases models take more time to write.

However, we also came across circumstances where some have suggested that the Use Case models would be compatible with Epics [6], which are large User Stories that contains collections of smaller User Stories.

1.1 Motivation – Why are Modular Transformations needed at all?

Two different surveys, one covering Sri Lankan, Singaporean and Australian IT Industries, [8] and another (done by us) concentrating exclusively on the Sri Lankan software development organizations, showed that some organizations use both UML as well as Non-UML Modelling methodologies, (even during Agile processes) to help design their software. Redundancy and unnecessary waste of time could be avoided, if other types of diagrams could be derived from one basic type.

The Survey that was covering Sri Lankan, Singaporean and Australian IT Industries was done by interviewing Project Managers, Business Analysts, Systems Analysts, Quality Analysts, Software Architects and Programmers. According to the results of the survey, 27% say that they also use non OO Modelling techniques for their projects. Our survey which was done for Industry as well as for Government Organizations who are involved with Software Design Development. Sixty four Organizations were involved for this purpose. Some questions asked with the % responses are given below.

Q1. Do you use Agile Methods / Processes?

Table 1 : Usage of Agile Methods

Yes	54	84.4%
No	10	15.6%

According to Table 1, 54% of the respondents use agile methods. Out of the 54% who uses Agile Methods, 51.9% say they use the popular Scrum Process where as 33.3% says they use a hybrid version (Mixture of different processes)

Q2. What modeling techniques do you use for software engineering?

Table 2: Modelling Techniques used for Software Engineering

a) UML Diagrams	13	20.3%
b) Mixture of UML and other modeling techniques	36	56.3%
c) Non-UML modeling techniques	11	17.2%
d) Do Not use modeling techniques	4	6.3%

According to Table 2 the majority (56.3%) seems to be using a mixture of UML and other modelling techniques for their projects. Out of the people who have responded for options b) and c), 79.6% uses Entity Relationship Diagrams (ERD) whereas 55.1% are using User Stories for their projects.

Q3. What UML Diagrams do you use for modelling? (From those who have answered Q2 (a) or (b))

Table 3 : Diagram wise usage of UML Diagram

Use Case Diagrams	32	86.50%
Class Diagrams	32	86.50%
Sequence Diagrams	21	56.80%
State Diagrams	13	35.10%
Activity Diagrams	20	54.10%
Other Diagrams	18	54.00%

According to Table 3, out of the respondents who are using UML diagrams, majority seems are using Class (86.5%) and Use Case Diagrams (86.5%) for their projects. For our Modular Transformation project we have given priority for the following diagrams and techniques for the initial stage based on the responses received for the questionnaire.

- 1) Use Case – from Q3. – 86.5% uses Use Case diagrams
- 2) *Use Stories* Q2 – from options b)and c) , 55.1% uses User Stories
- 3) ERD – Q2 – from options b)and c) , 79.6% uses ERD diagrams
- 4) Class - from Q3. – 86.5% uses Class diagrams

In this paper we focus on the User Stories and Use Cases. In the case of Modular Transformations between User Stories and Use Case models, the Uexceler tool contained within the Visual Paradigm software [9] shows an attempt to link up the usage of Use Cases models and User Stories with each other.

2.0 Methodology and Results

A significant fact which first needs to be highlighted is that Use Case models and User Stories are both very important techniques in the Requirement Gathering Process of software development. Whether your project requires a User Story, Use Case, or both, will depend on the project, the available collaboration, the level of formality, and the upfront research required. “Some have found success in a hybrid, such as a highly detailed User

Story, while others find the User Story as an important launching off point for the more detailed Use Case.” [11]

According to Segue [11], they tend to employ Use Cases predominantly with government projects that have more stringent documentation requirements. For smaller projects having short duration they tend to lean towards User Stories. The many-years debate shows not so much that one is better than the other but that each can be applicable in varying degrees from project to project. Another fact is that, although there are no distinct phases such as requirement gathering phase or design phase in agile processes, the need for efficient and effective requirement gathering is there, since the Industry develops software for clients / customers who have a low software development literacy.

2.1 Identifying Actors and Use Cases from User Stories

When looking at User Stories and Use Cases, one feature that would strike one would be that the role in a user story is similar to an actor of a use case model, while the goal or desire in a user story is similar to a use case. [7]

However, instead of being directly related to User Stories, Use Case models maybe related to them through Epics. Epics are large User Stories which could be further decomposed into many smaller User Stories [7][10] for the purpose of streamlining Software Development. An Epic is a larger user story which is too big to implement in a single iteration and therefore need to be disaggregated into smaller user stories at some point. [12]

We realized that even if User Stories may not be directly compatible with Use Cases, Epics which are larger and broader in scope maybe so.

We also examined some examples of Epics given on the internet, and studied how their relevant authors have broken them down into smaller User Stories. For example, the following Epic

“As a hotel operator, I want to set the optimal rate for rooms in my hotel.” could be broken up into two smaller user stories as follows;

“As a hotel operator, I want to set the optimal rate for rooms based on prior year pricing.” and

“As a hotel operator, I want to set the optimal rate for rooms based on what hotels comparable to mine are charging.”.

When comparing Use Cases (eg. OrderFood, SearchItems, Login) and User Stories (eg. I want to < select food / enter search key >) it appears that User Stories are much simpler tasks than Use Cases.

However, when we consider Epics, (As a customer I want to < order food / search items >, so that) it is apparent that there is a similarity between them and Use Case models.

Roles and Epics can be used to generate Use Case models.

2.2 Experiment conducted to determine the Justification of the Compatibility between Epics and Use Case Models

An Experiment was conducted with a sample of up to 160 Computer Science under graduate students. Students were given a Case Study and asked to develop User Stories followed by drawing the Use Case Model using the case study.

We found that 148 students proposed in a positive way that User Stories could be transformed into Use Case models (See Table 4) while 12 students expressed a neutral opinion. It was significant and interesting that none of the students replied negatively with respect to transforming user Stories into Use Cases.

Table 4 –Compatibility between Epics and Use Cases

Nature of Opinion	No. of Students
Positive Opinion	148
Neutral Opinion	12
Negative Opinion	0

Identifying Include relationships from user stories

We found that 70 students proposed in a positive way that common User Stories found in Epics can be considered for *include* relationships in the Use Case model whereas 90 have expressed neutral opinion and none has responded negatively. (See Table 5).

Table 5 –Include Relationship vs User Stories

Nature of Opinion	No. of Students
Positive Opinion	70
Neutral Opinion	90
Negative Opinion	0

Based on the Experiment the following example illustrates the connection between User Stories, Epics and Use Cases. Say in a library, when a member wants to borrow a book, the librarian needs to do the following:

- Check whether the member is valid (registered member)
- Check whether the member has overdue books
- Check whether the copy to be borrowed is a borrowable copy (not reference)

Users Stories, Epics and the Use Case diagram segment for the above scenario are given below in Figure 1.

User Stories :-

- As a Librarian I want to check the validity of the member so that I can issue the book*
- As a Librarian I want to check for overdue books so that I can issue the book*
- As a Librarian I want to check for type of copy so that so that I can issue the book*

Epic :-

- As a Librarian I want to Issue books for members so that members can borrow them*

Use Case Diagram :



Figure 1 - Users Stories, Epics and the corresponding Use Case diagram segment

Based on the analysis done after the experiment we observed some connection between *include* relationship and the user stories.

The example given below illustrates the connection between User Stories, Epics and the *include* relationship in a Use Case model.

Say in a library, when a member wants to reserve a book, the librarian needs to do the following:

- Check whether the member is valid (registered member)
- Check whether the member has overdue books

The Figure 2 illustrates the User stories, Epics and Use Case model with an *include* relationship.

User Stories :-

- As a Librarian I want to check the validity of the member so that I can reserve the book*
- As a Librarian I want to check for overdue books so that I can reserve the book*

Epic :-

- As a Librarian I want to Reserve a book for a member so that member can borrow it later*

Use Case Diagram

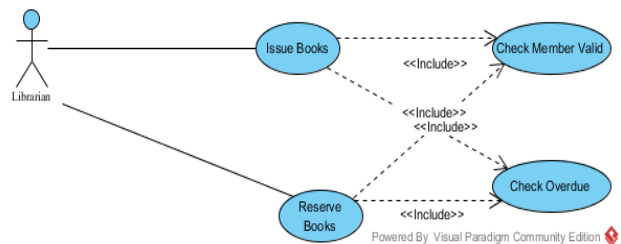


Figure 2 - User stories, Epics and Use Case model with an *include* relationship.

Identifying *Extend* relationships from user stories

We found that only 8 students proposed in a positive way that conditional User Stories (if provided) in Epics can be considered for *extend* relationships in the Use Case model whereas 148 have expressed neutral opinion and 04 have responded negatively. (See Table 5).

Table 5 –Include Relationship vs User Stories

Nature of Opinion	No. of Students
Positive Opinion	08
Neutral Opinion	148
Negative Opinion	04

Based on the positive responses we can suggest the following.

Say in a library, if the librarian found overdue books he needs to Issue a fine.

The Figure 3 illustrates the User stories, Epics and Use Case model with an *Extend* relationship as we suggest.

User Stories :-

As a Librarian I want to check the validity of the member so that I can issue the book

As a Librarian I want to check for overdue books so that I can issue the book

As a Librarian I want to issue a fine so that I can issue the book after the payment is settled

Epic :-

As a Librarian I want to Issue books for members so that members can borrow them

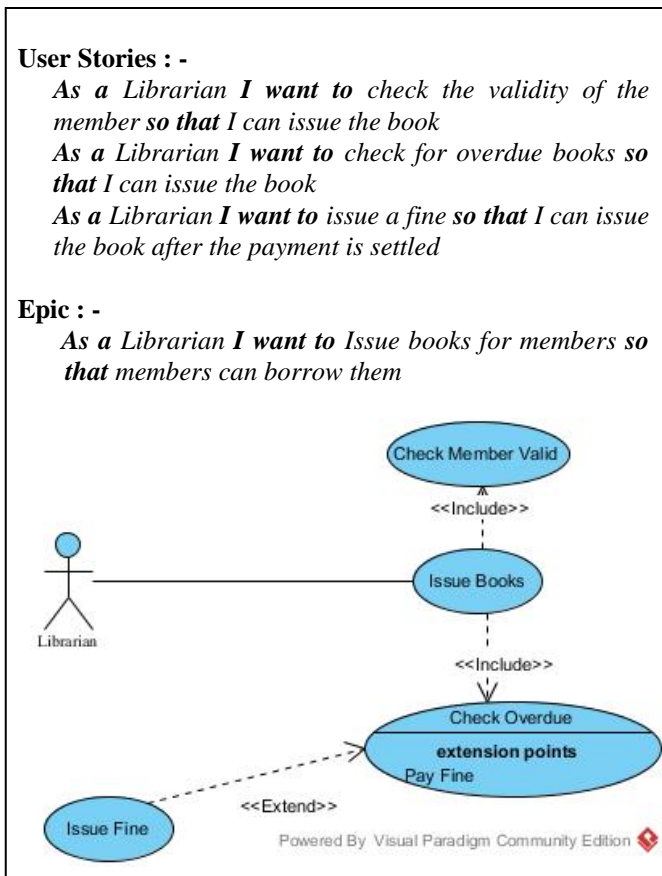


Figure 3 - User stories, Epics and Use Case model with an *extend* relationship.

Discussion and Conclusion

A feature which we found which is shared by both Epics and Use Case models, is that they are both large in scope. “Search Items”, which could be a Use Case, could just as easily be a goal / desire in an Epic and it is quite large in scope. However “Enter Search String”, which could be a sub-task of the Epic “Search Items”. A simpler User Story is smaller in scope compared to both the Epic and the Use Case. Therefore it can be seen that User Stories by contrast, are smaller compared with both Use Case models and Epics, and that Use Case models maybe indirectly related to User Stories due to their similarity with Epics. Further the use cases for the *include* relationships can be identified by looking at the similar user stories in Epics. Finally our opinion is that if the User stories can be written with conditions identifying the use cases for the *extend* relationships is also possible.

Our next step is to design a prototype of a software to transform Roles and Goals / Desires of Epics into Actors and Use Cases of Use Case models.

References

- [1] Fries, T.P. (2006), “A Framework for Transforming Structured Analysis and Design Artifacts to UML”, *The 24th annual conference on Design of communication*.
- [2] Jilani A.A.A., Usman M., Nadeem A., Malik Z.I. and Halim Z., (2011), “Comparative Study on DFD to UML Diagrams Transformations,” *World of Computer Science and Information Technology Journal (WCSIT)*, pp, 10-16.
- [3] Tilakaratna, P. and Rajapakse, J. (2011), Forward Engineering the Object Oriented Analysis and Design, *The 5th Malaysian Conference in Software Engineering*, 107-112.
- [4] Tilakaratna P., and Rajapakse J., (2012) “Ontological Framework for Object-Oriented Analysis and Design,” *American Journal of Engineering and Applied Sciences* 5 (3), 251-260.
- [5] M.Chacon (2015, October 16). [Online]. Available: <https://www.scrumalliance.org/community/articles/2015/october/user-stories-vs-use-cases>
- [6] Jacobson, I. and Spence I., (2010) “Using Use Cases on Agile Projects”, 10.
- [7] How can we map use cases to stories?: <https://agilefaq.wordpress.com/2008/06/18/how-can-we-map-use-cases-to-stories/>
- [8] Tilakaratna, J.M.P.P., (2016) “Development and Evaluation of an Ontologically Complete and Clear Object-Oriented Conceptual Modelling Grammar”, PhD Thesis, 251-256.
- [9] Hong Kong Institute of Vocational Education. (2016, February 3). [Online]. Available: <https://www.visual->

paradigm.com/support/documents/vpuserguide/2607/2608/83675_overviewofue.html

- [10] Atlassian Documentation (2016). [Online]. Available: <https://confluence.atlassian.com/agile/jira-agile-user-s-guide/working-with-epics/completing-an-epic>
- [11] User stories vs use cases, agile development, Segue Technologies [2016]
<http://www.seguetech.com/user-stories-vs-use-cases-pros-cons-agile-development/>
- [12] User Stories: An Agile Introduction [2016]
<http://www.agilemodeling.com/artifacts/userStory.htm>



R. S. Madanayake received his BSc in Biological Science (2001) and his MSc in Information Technology (2009) both from the University of Colombo, Sri Lanka, and is currently reading for an MPhil at the University of Colombo, Sri Lanka. He has 02 fullpaper publications and 02 extended abstracts to his credit, all 04 of which are in the field of Modular Transformations involving Agile, UML and Structured Analysis and Design Techniques. His other interests include how Ontologies and Eastern Four-valued Logic fit into the field of Computer Science.



G. Kapila .A. Dias received his BSc in Physical Science (1982) from the University of Colombo, Sri Lanka, Postgraduate Diploma in Computer Studies from University of Essex, UK (1986), and MPhil in Computer Science from University of Wales, UK (1995). He is currently a Senior Lecturer at the University of Colombo School of Computing (UCSC). He is a member of the Computer Society of Sri Lanka, Sri Lanka Association for Advancement of Science (SLASS) and a member of the ACM the world's largest educational and scientific computing society. He received the winner award for his 'Travel Meter' project in eSwarbhimani 2016, Sri Lanka under Culture & Tourism Category. He also won the Bronze award under Education & Training category in NBQSA Sri Lanka-National Best Quality Software Award -Oct 2015 for K8-Flight Simulation project. He has more than 50 peer reviewed publications to his credit. His research interests are computer aided software engineering, multimedia for education, modeling and simulation, artificial neural networks and model driven engineering.



Nihal D. Kodikara is a Professor attached to the University of Colombo School of Computing (UCSC). His research interests include Image Processing, Computer Vision, Computer Graphics and Software Engineering. Prof. Kodikara obtained his B.Sc. from University of Colombo and M.Sc. and Ph.D. in Computer Science from the University of Manchester, U.K. He has received Colombo Plan Technical Award in 1986 for Ph.D. studies and Commonwealth Fellowship Award in 1997. He has over 35 years of teaching and research experience. He is presently the Head of the Modelling and Simulation Research Group of the UCSC. He has more than 70 peer reviewed publications to his credit.