# User's Manual as a Requirements Specification

**Daniel M. Berry, 1991, 1998, 2002, and 2003 with help from K. Daudjee, J. Dong, I. Fainchtein, M.A. Nelson, T. Nelson, & L. Ou**

**2004**

# A Case Study of Serious RE

**A Master's student of mine, Lihua Ou, did a case study of writing requirements specification in the form of a user's manual [Berry *et al* (Ou) 2004].** **2002**

**It was *very* successful in that I got a piece of software that I wanted, it was implemented well, it does what I want it to do, and there is a well-written manual that describes the software's behavior completely.**

# A Case Study, Cont'd

**Along the way, it ended up being also a case study in just having a serious requirements process, in which implementation did not begin, and was in fact *delayed*, until the requirements were completely worked out and specified satisfactorily.**

# The Software

**The software was a WYSIWYG, direct manipulation picture drawing program, WD-PIC, based on the batch picture drawing language PIC, a TROFF preprocessor.**

**Lihua Ou's assignment was to produce a first production-quality version of WD-PIC as her master's thesis project.**

# Ou's Professional Background

**Prior to coming to graduate school, Ou had built other systems in industrial jobs, mainly in commerce.**

**She had followed the traditional waterfall model, with its traditional heavy weight SRS.**

**She had made effective use of libraries to simplify development of applications.**

# Ou's Input

**Ou was to look at all previous prototypes and UMs as specifications.**

**She was to filter these and scope them to first release of a production quality version of WD-PIC running on Sun UNIX systems.**

# Ou's Assignment

**Ou was to write a specification of WD-PIC in the form of a UM.**

**This UM was**

1. **to describe all features as desired by the customer, and**
2. **to be accepted as complete by the customer,**

**before beginning design or implementation.**

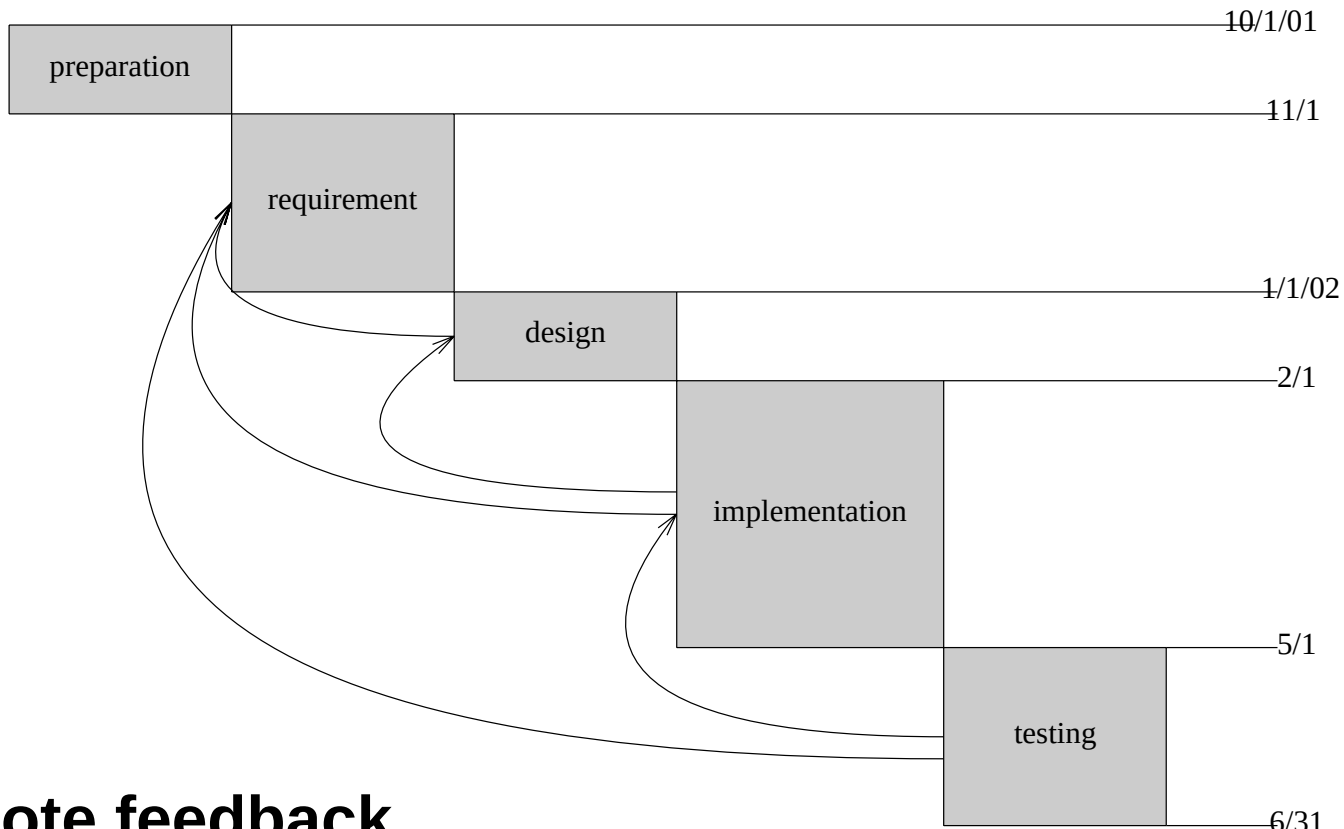
# Ou's Assignment, Cont'd

**Once implementation started, whenever new requirements were discovered, the UM had to be modified to capture new requirements.**

**In the end, the UM was to describe the program as delivered.**

# Project Plan

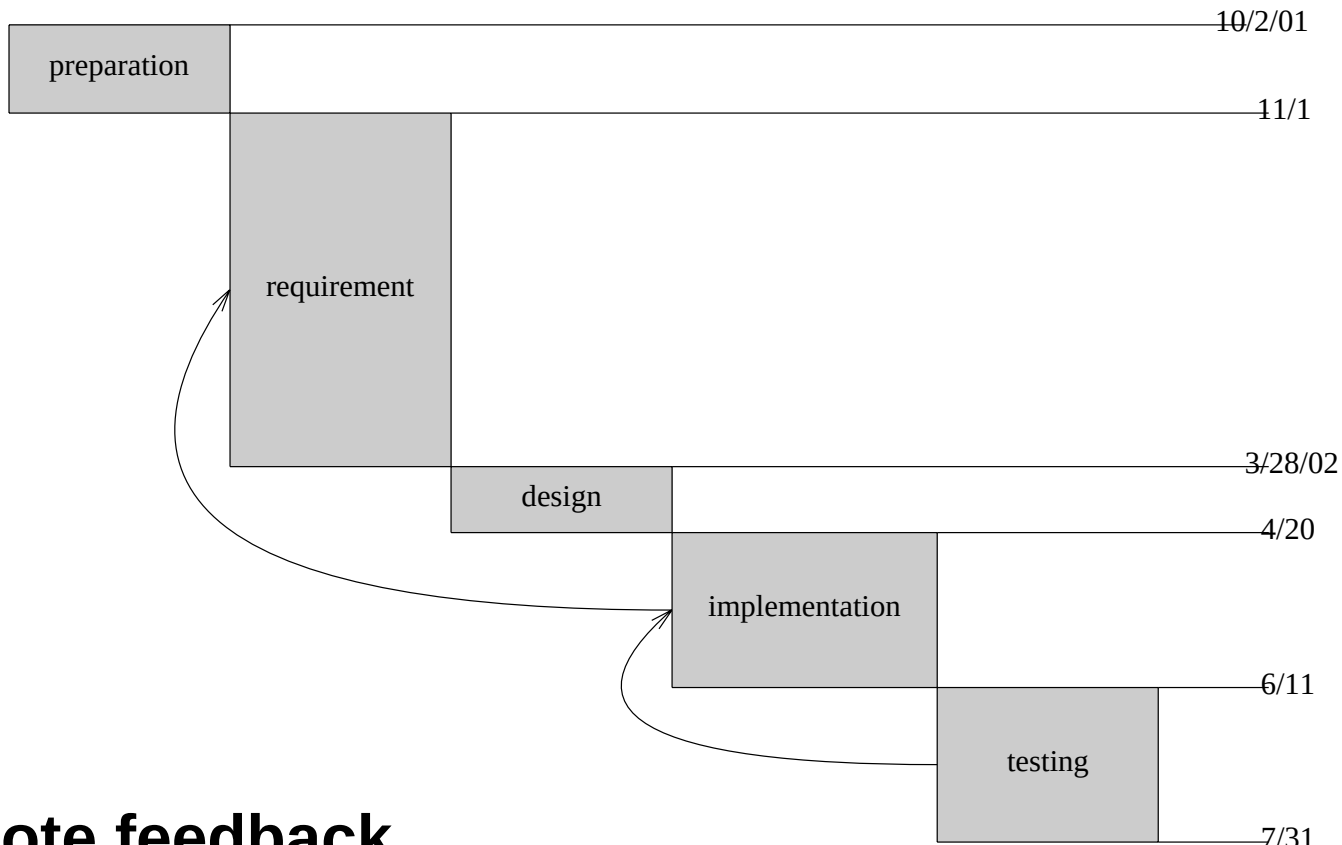
| Duration in months | Step |
|---|---|
| 1 | Preparation |
| 2 | Requirements specification |
| 4 | Implementation |
| 2 | Testing |
| 1 | Buffer (probably more implementation and testing) |
| 10 | Total planned |




# Note feedback

# Actual Schedule

| Duration in months | Step |
|---|---|
| 1 | Preparation |
| 4.9 | Writing of user's manual = reqs spec, 11 versions |
| .7 | Design including planning for maximum reuse of PIC code and JAVA library |
| 1.7 | Implementation including module testing and 3 manual revisions |
| 1.7 | Integration testing including 1 manual revision and implementation changes |
| 10 | Total actual |




# Note feedback

# What Happened?

**While detailed plan was not followed, total project time was as planned.**

**Also, Ou produced two implementations for the price of one, for:**

- **(planned) Sun with UNIX and**
- **(unplanned) PC with Windows 2000**

# Surprise

**Ou was more surprised than Berry that she finished on time.**

**Berry had a lot of faith in the power of good RE to reduce implementation effort.**

**Adding to Ou's surprise was that the requirements phase took nearly 5 months instead of 2 months; the schedule had slipped 3 months out of 10, what appeared to be *way* beyond recovery.**

# Then and ...

**Ou's long projected implementation and testing times and the 1 month buffer indicate that she expected implementation to be slowed by discovery of new requirements that necessitate major rewriting and restructuring.**

# Then and Now

**This time, only minor rewriting and no restructuring.**

**Thus instead of 2 months specifying and 7 months implementing and testing,**

**she spent 5 months specifying and only 4 months implementing and testing.**

# Why?

**By spending 3 additional months writing a specification that satisfied a particularly hard-nosed customer who insisted that the manual convince him that the product already existed,**

**Ou produced a specification that**

- **had very few errors and**
- **that was very straightforwardly implemented.**

# The Errors

**Almost all errors found by testing were relatively minor, easy-to-fix implementation errors.**

**The two requirement errors were relatively low level and detailed.**

**They involved subfeatures in a way that required only very local changes to both the UM and the code.**

# What Helped?

**All exceptional and variant cases had been worked out and described in the UM.**

**Thus, very little of the traditional**

- **implementation-time fleshing out of exceptional and variant cases and**
- **implementation-time subconscious RE.**

# Test Cases

**The manual's scenarios, including exceptions and variants turned out to be a complete set of black box test cases.**

**Tests were so effective that, to our surprise, ...**

**scenarios not described in the UM, but which were logical extensions and combinations of those of the UM worked the first time!**

**The features composed orthogonally without a hitch!**

# Satisfied Customer

**Berry found Ou's implementation to be production quality and is happily using it in his own work.**