

Intelligent Software Engineering

Requirements Engineering

Zhilei Ren



Dalian University of Technology

August 8, 2025



Definition of Requirements Engineering

Requirements engineering is an interdisciplinary function that mediates between the domains of the acquirer and supplier or developer to establish and maintain the requirements to be met by the system, software or service of interest. Requirements engineering is concerned with discovering, eliciting, developing, analyzing, verifying (including verification methods and strategy), validating, communicating, documenting and managing requirements¹.

¹ISO/IEC/IEEE 29148 Systems and software engineering —Life cycle processes – Requirements engineering ▶



As Proposed by the Project Sponsor



As Specified in the Project Request



As Designed by the Senior Systems Analyst



As Produced by the Programmers



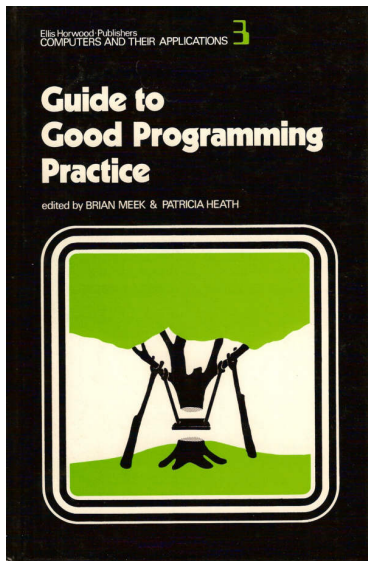
As Installed at the User's Site



What The User Wanted



Guide to Good Programming Practice, 1979



Research Topics in Requirements Engineering

- ① Requirements Classification
- ② Requirements Prioritization
- ③ Feature Model Optimization
- ④ Prototype Generation



Techniques for Requirements Engineering Research



Next Release Problem

Given:

- A set of software requirements $R = \{r_1, r_2, \dots, r_n\}$,
- A set of customers $C = \{c_1, c_2, \dots, c_m\}$,
- Each customer $c_j \in C$ requests a subset of requirements $R_j \subseteq R$ and provides a profit $p_j > 0$ if all requirements in R_j are satisfied,
- Each requirement $r_i \in R$ has an associated cost $\text{cost}(r_i) > 0$,
- A total available budget $B > 0$.

The goal is to select a subset of requirements $R' \subseteq R$ such that:

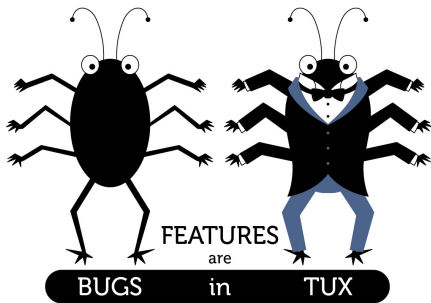
- 1 The total cost does not exceed the budget: $\sum_{r_i \in R'} \text{cost}(r_i) \leq B$,
- 2 The total profit is maximized:

$$\max_{\substack{R' \subseteq R \\ c_j \in C \\ R_j \subseteq R'}} \sum p_j.$$

This is known as the **Next Release Problem (NRP)** and is a well-known NP-hard problem in requirements engineering and software release



bug or feature?




The First Computer Bug

Photo # NH 96566-KN (Color) First Computer "Bug", 1947

9/9

0800 Antan started
 1000 " stopped - antan ✓ { 1.2700 9.037 897 025
 13.02 (032) MP - MC 1.926 9.037 896 995 convt
 (033) PRO 2 2.13047645 9.615925059(-2)
 convt 2.13047645
 Relays 6-2 in 033 failed special speed test
 in relay " 11,000 test. Relay
3145
Relay 3370

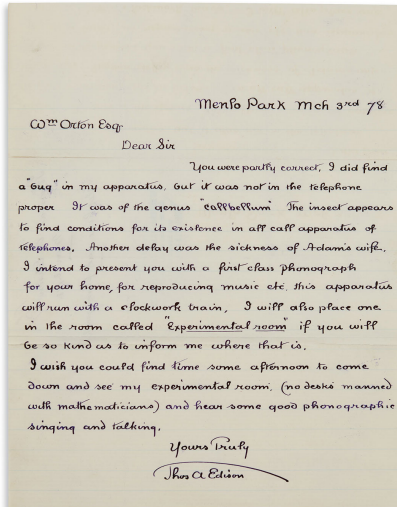
1100 Started Cosine Tape (Sine check)
 1525 Started Multi-Adder Test.

1545  Relay #70 Panel F
 (Math) in relay.

First actual case of bug being found.
 1630 Antan started.
 1700 closed down.



The First Computer Bug



Combo

Combos were a design accident; lead producer Noritaka Funamizu noticed that extra strikes were possible during a bug check on the car-smashing bonus stage. He thought that the timing required was too difficult to make it a useful game feature, but left it in as a hidden one².



²[https://en.wikipedia.org/wiki/Combo_\(video_games\)](https://en.wikipedia.org/wiki/Combo_(video_games))



彩蛋

