Estimating Software Requirements

Speaker: Yong Xia

Seminar on Software Cost Estimation

January, 14th 2003

Motivation

- Requirements are important
- Most researches forget two points:
  - Quantification of requirements sizes, schedules, effort, and costs
  - Quantification of requirements errors and defect-removal efficiency

Structure

- Background
- Requirements Estimates
- Requirements Errors
- Evaluating Combinations of Requirements Factors

Background

- Function Points (a synthetic metric)
  - Five attributes of software system:

    Input, Output Inquiries, Logical Files, Interface
- Defect-Prevention technologies
- Type of Software
- Positive and Negative Requirements Adjustment Factors

Defect-Prevention Technologies

- Joint Application Design (JAD)

A method for developing software requirements under which user representatives and development representatives work together to produce a joint requirements specification that both sides agree to

- Quality Function Deployment (QFD)

QFD operates in a fashion similar to JAD, but focuses on the quality needs of the application rather than on general requirements

- Prototype
- Use cases
- Change-Control Boards

CCBs are not exactly a technology, but rather a group of managers, client representatives, and technical personnel who meet and decide which changes should be accepted or rejected

Types of software

- End-User Software
- Management Information System

Software requirements are usually derived from users or the users’ authorized representatives

- Outsourced Projects
- System Software
- Commercial Software
- Military Software

Positive requirements factors

- High client experience levels
- High staff experience levels
- Joint application design (JAD)
- Prototyping
- Quality function deployment (QFD)
- Use cases
- Requirements inspections
- Reusable requirements (patterns or frameworks)
- Requirements derived from similar projects
- Requirements derived from competitive projects
- Effective requirements representation methods

Negative Requirements Factors

- Inexperienced clients
- Inexperienced development team
- Novel applications with many new features
- Requirements creep of more than 3 percent per month
- Ineffective or casual requirements-gathering process
- Failure to prototype any part of the application
- Failure to review or inspect the requirements
- No reusable requirements

Structure

- Background
- Requirements Estimates
- Requirements Errors
- Evaluating Combinations of Requirements Factors

Requirements Estimates

An example (c.f. Table 1)
with a 1500-function point system

- 375 pages requirements
- 9 staff months
- 3 calendar months
- 1500 potential defects
- 450 high-severity requirements defects (30%)
- 375 delivered requirements defects
- 112 high-severity latent defects

Requirements Productivity Rates:

Ranges in Requirements Productivity Rates by Class of Software

Software class	Requirements productivity, FP/staff month	Requirements productivity, staff hours/FP
End user	1000	0.128
Commercial	200	0.640
Small MIS	175	0.750
Large MIS	75	1.710
Outsource	90	1.422
Systems	75	1.710
Military	35	3.657

Other attributes

- Performed by
- Formal methodologies
- Requirements tools
- Defect prevention methods

Structure

- Background
- Requirements Estimates
- Requirements Errors
- Evaluating Combinations of Requirements Factors

Requirements errors:

Requirements Defects and Other Categories

Defect origins	Total defects per FP	High-severity defects per FP
Requirements	1.00	0.30
Design	1.25	0.50
Code	1.75	0.25
Documentation	0.60	0.10
Bad fixes	0.40	0.15
Total	5.00	1.30

Requirements change:

Monthly Growth Rate of Software Creeping Requirements

Software type	Monthly rate of requirements change, %
Contract or outsource software	1.0
Information systems software	1.5
System software	2.0
Military software	2.0
Commercial software	3.5

Structure

- Background
- Requirements Estimates
- Requirements Errors
- Evaluating Combinations of Requirements Factors

Evaluating Combinations of Requirements Factors

4 factors:

- The use of or failure to use prototypes
- The use of or failure to use joint application design (JAD)
- The use of or failure to use formal requirements inspections
- The presence or absence of experienced staff familiar with the application type

Sixteen Permutations of Software Requirements Technologies

- SPR assumes fairly complex applications of at least 1,000 function points or 125,000 C statements in size.
- For smaller projects, requirements defects and rates of change would be less.
- For large systems in excess of 10, 000 function points or 1,125,000 C statements, requirements errors would be larger and removal efficiency would be lower.

Discussion

- Questions?
- Personal Opinions