IPA Software Engineering Center
Information-technology Promotion Agency, Japan

PSAM9
“Testing Framework for embedded software based on software safety requirement assessment.”

Masayuki HIRAYAMA, Satomi YOSHIZAWA, Yutaka UKON

Software Engineering Center
Information-technology Promotion Agency, Japan
Yutaka UKON

Introduction - 1

Contemporary State in Embedded Systems

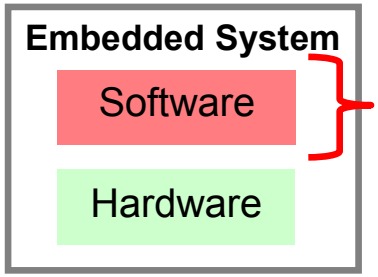
- There are various embedded systems in our daily life.
  - Consumer products : Cellular phone, Digital Appliances…etc…
  - Social infrastructure systems : Power plant, Aircraft…etc..

Even if there is just a difference in the type of a system, system troubles caused by various embedded system give serious influence on our daily.

- Background of there software developments.
  - Since the development period of software **is restricted**, the **time** which can be spent on a **review** or **a test is also restricted**.

Introduction - 2

How to assure the safety of the software



Tend to be realize various function by the software

- In order to develop safety software, building high quality into embedded software is required.

For assuring software quality by document review and system test

- Need to spent time on a review or test.
- Need technique of reviewing or testing with high efficiency.

Purpose of our research

Establish an effective implementing framework for embedded software's quality.

Problems - 1

Required safety level

- Impact of trouble in embedded software changes with types of software or system.

Home appliances		**Power plant**
May not cause human loss	⟷	**May cause wide range of human loss**

Required Safety level
is
quite **Different**

Like these problems, we should prepare suitable indexes for embedded software depending on the safety level.

Problems - 2

Software quality implementation technique

Each engineer

- **To develop software with high quality**
  - Detail review for software design
  - Rigorous testing with high testing coverage of source code

Clearly Defined

- **Standard value**
  - Review effort
  - Test effort
  - Number of test items
  - …

Target value

Not Defined

The suitable indexes that considers system safety is unpracticed.

Outline of ESSI

- ESSI : Embedded Software Safety Implementation technique

The ESSI consists three steps the following;

Step-1 :Calculation of assumption amount of damage for embedded software trouble

→ The assumption amount of damage is calculated from number of software users, degree of damage and occurrence ratio of damages.

Step-2 :
Categorize of software types

→ There are three types.
1. Highly critical
2. Critical
3. Normal

Step-3 :
Quality target value setting

→ Target values of the quality indexes .
1. Ratio of test effort
2. Ratio of review effort

＊ Adjusted and defined

Step 1 - TAD calculation

Meaning of each parameter is as follows.

CDI : Coefficient of Damage Impact

DRI : Damage Ratio of Incident

NU : Number of users

PMS : Period of Non service

HAD calculation . . Amount of assumed Human Damage

EAD calculation . . Amount of assumed Economic Damage

TAD calculation . . Total Amount of assumed Damage

＊TAD = HAD+EAD

HAD calculation

- HAD is an index which expresses the amount of human damage for embedded software incident

- HAD=sum (NU×DRI×CDI)
  - NU : Number of users
  - DRI : Damage occurrence Ratio of Incident
  - CDI : Coefficient of Damage Impact

- CDI for Human damage calculation
  - Categorizing the human impact for users into three category: Type-A, B, C
  - CDI is defined according to the type

Type of Human damage	Meanings	CDI (JPY)
A	Human life loss	\100,000,000.-
B	Serious injured	\5,000,000.-
C	Slight injured	\100,000.-

EAD calculation

- EAD is an index which expresses the amount of economic damage for embedded software incident
- EAD=sum (NU×PNS×DRI×CDI)
  - NU : Number of users
  - PNS : Period of Non service
  - DRI : Damage occurrence Ratio of Incident
  - CDI : Coefficient of damage impact

- CDI for Economic damage calculation
  - Categorizing the economic impact for users into six levels
  - CDI is defined according to the type

Type of Economic damage	CDI
A	\100,000,000.-
B	\5,000,000.-
C	\500,000.-
D	\50,000.-
E	\5,000.-
F	\500.-

Step 2 - Software categorization

Categorize of embedded software

- By referring TAD (Total Amount of assumed Damage) from software safety viewpoint, embedded software is categorized into three types

Quality Type	TAD
Highly Critical	\10,000,000,000
Critical	\1,000,000,000
Normal	\10,000,000

Step 3 - Quality target value setting

Show by a list for each Values

- According to the evaluation of quality type in step-2, quality target values system are defined.
- As for the quality implementation index, in consideration of controlling the degree of review sufficiency, testing coverage, an index and a desired value as shown below are prepared

			Highly Critical	Critical	Normal
TD	Test item density	Number of test item / KLOC	80	50	20
CF	Convergence of fault	Convergence of fault management curve	100	97	94
RT	Ratio of test effort	Test effort / Total development effort	5	3	1
RDR	Ratio of design review effort	Design review effort / Total development effort	5	3	1
RCR	Ratio of code review effort	Code review effort / Total development effort	2.5	1.5	0.5

Simple case study (1)

Case 1 : Personal printer

- Outline
  - Number of user: 500,000. (NU=500,000)
  - Software trouble shooting is completed within two days (PNS=2.0)
- Step-1:Calculation of TAD
  - HAD=0
  - EAD:
    - CDI: if the ordinary person depends for 5000 yen work per day on a personal printer, CDI=\5,000.- (Type-E)
    - DRI: if damage occurs to 10 percent of all users, and one percent of all user request their product's maintenance, DRI=0.1×0.01
  - EAD =sum (NU×PNS×CDI×DRI)
    = 500,000 × 2.0 × \5,000.- × 0.1× 0.01
    = \5,000,000.-
  - TAD = HAD + EAD = \5,000,000.-

Simple case study – case 1 (cont')

- Step-2 : Software Categorization
  - TAD =\5,000,000.-
  - Quality Type: Normal

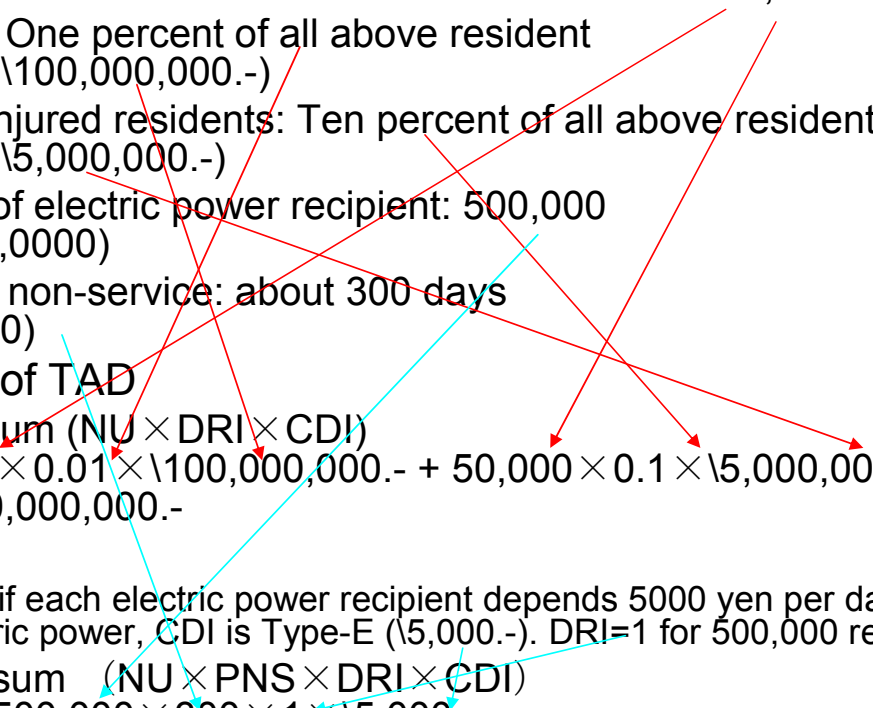
Quality Type	TAD
Highly Critical	\10,000,000,000
Critical	\1,000,000,000
Normal	\10,000,000

- Step-3: Quality implementation index
  - TD=20, CF=94, RT=1, RDR=1, RCR=0.5

			Highly Critical	Critical	Normal
TD	Test item density	Number of test item / KLOC	80	50	20
CF	Convergence of fault	Convergence of fault management curve	100	97	94
RT	Ratio of test effort	Test effort / Total development effort	5	3	1
RDR	Ratio of design review effort	Design review effort / Total development effort	5	3	1
RCR	Ratio of code review effort	Code review effort / Total development effort	2.5	1.5	0.5

Simple case study – 2

- Electric power plant controller
  - Outline of the assumed trouble
    - Number of the residents who live on the outskirts: 50,000
    - Life loss: One percent of all above resident (Type-A, \100,000,000.-)
    - Serious injured residents: Ten percent of all above resident (Type-B, \5,000,000.-)
    - Number of electric power recipient: 500,000 (NU=500,0000)
    - Period of non-service: about 300 days (PNS=300)
  - Calculation of TAD
    - HAD =sum (NU×DRI×CDI)
      = 50,000×0.01×\100,000,000.- + 50,000×0.1×\5,000,000.-
      = \75,000,000,000.-
    - EAD
      - CDI: if each electric power recipient depends 5000 yen per day on the electric power, CDI is Type-E (\5,000.-). DRI=1 for 500,000 recipient.
    - EAD = sum (NU×PNS×DRI×CDI)
      = 500,000×300×1×\5,000.-
      = \750,000,000,000.-
    - TAD = HAD + EAD = \825,000,000,000.-



Simple case study – case 2 (cont')

- Step-2: Software Categorization
  - TAD = \825,000,000,000.-
  - Quality Type: Highly Critical

Quality Type	TAD
Highly Critical	\10,000,000,000
Critical	\1,000,000,000
Normal	\10,000,000

- Step-3:Quality implementation index
  - TD=80, CF=100, RT=5, RDR=5, RCR=2.5

			Highly Critical	Critical	Normal
TD	Test item density	Number of test item / KLOC	80	50	20
CF	Convergence of fault	Convergence of fault management curve	100	97	94
RT	Ratio of test effort	Test effort / Total development effort	5	3	1
RDR	Ratio of design review effort	Design review effort / Total development effort	5	3	1
RCR	Ratio of code review effort	Code review effort / Total development effort	2.5	1.5	0.5

Conclusion

- ESSI
  - Embedded Software Safety Implementation technique
  - Amount of total damage of the software is calculated by Evaluating,
    - The number of users (NU)
    - Type of damage for software trouble (CDI)
    - Amount of damage (TAD = HAD + EAD, includes PNS)
    - Occurrence ratio (DRI)
  - Each software is categorized by referring to the amount of damage
  - Software testing effort or software review effort is optimized according to the software required safety level
  - We will be investigating and evaluating the adaptability of ESSI by applying various type of embedded software
    - Personal printer, Electric power plant and so on
  - We would like to improve ESSI to more sophisticated method

Q&A