

Requirements Engineering

Stuart Anderson

Room 1610, JCMB, KB

0131 650 5191, soa@inf.ed.ac.uk

SEOC1 – 2003-4

Reading/Activity

- Please read pages 9-24 of the SWEBOK, this provides a basic outline of the requirements process.
- Please read pages 25-46 of the UML book in preparation for the next lecture.
- Try running argo on one of the DICE machines (just type argo in a shell window).

SEOC1 – 2003-4

Volunteer Bank (VolBank)

1. To develop a system that will handle the registration of volunteers and the depositing of their time. To record:
 - i. The details of volunteers, contact details, skills and needs
 - ii. The time that each volunteer deposits in the system
 - iii. To transfer from the web server details of volunteers and the time they are depositing.
2. To handle recording of opportunities for voluntary activity:
 - i. Details of voluntary organisations
 - ii. Needs of voluntary organisations
 - iii. Needs of individuals (inc volunteers) for help

SEOC1 – 2003-4

VolBank continued

3. To match volunteers with people or organisations that need their skills:
 - i. Match volunteer with local opportunities
 - ii. Match local opportunity with a team of volunteers
 - iii. Record matches between volunteers and opportunities
 - iv. Notify volunteers of a match
 - v. Notify organisations of a match
 - vi. Record if agreement is reached from a particular match
4. To generate reports and statistics on volunteers, opportunities and time deposited.

SEOC1 – 2003-4

Software Requirements

- Main activities involved in Software Requirements engineering:
 - Elicitation: I identify sources; Elicit requirements
 - Analysis: Classify requirements; Model; Top-level architecture; Allocate requirements to components; Negotiate requirements
 - Specification: Requirements Definition Doc; Software Requirements Specification; Document Standards; Document Quality
 - Validation: Reviews; Prototypes; Modelling; Test definition
 - Management: Traceability; Attributes; Change/Evolution
- The pattern, sequence and interaction of these activities is orchestrated by a Requirements Engineering Process.

SEOC1 – 2003-4

VolBank: Elicitation

- Identify potential sources of requirements:
 - Goals (why the system is being developed): high level goal is to increase the amount of volunteer effort utilized by needy individuals and organisations – suggests possible requirements in measurement and monitoring
 - Domain Knowledge: not much relevant here but in some areas e.g. safety – hazard analysis; security – vulnerability and threat analysis
 - Stakeholders: at least: volunteers, organisations, system administrators, needy people, operator, maintenance, manager
 - Operating Environment: may be constrained by existing software and hardware in the office
 - Organisational Environment: legal issues of keeping personal data, safety issues in “matching”

SEOC1 – 2003-4

VolBank: Elicitation

- Approaches to eliciting requirements:
 - i. Interviews with stakeholders:
 - i. Operator identifies:
 - a. The need to change details when people move home
 - b. The need to manage disputes when a volunteer is unreliable, or does bad work
 - ii. Female Volunteer identifies: the need for security/assurance in contacting organisations, ...
 - iii. Management identifies number of hours volunteered per month above a given baseline as the key metric
 - ii. Scenarios: means to elicit the usual flow of work
 - iii. Prototypes: mock-up using paper or powerpoint or software
 - iv. Facilitated Meetings: professional group work
 - v. Observation: observing “real world” work.

SEOC1 – 2003-4

VolBank: Scenario: Failed Match

- Goal: to handle failure of a match
- Context: the volunteer and organisation have been matched and a date for a preliminary meeting established
- Resources: time for volunteer and organisation
- Actors: volunteer, operator, organisation
- Episodes: The volunteer arrives sees the job to be done and decides (s)he cannot do it, organisation contacts operator to cancel the match and reorganise.
- Exceptions: volunteer fails to show up

SEOC1 – 2003-4

VolBank: Requirements analysis

- Large volume of requirements information, need to analyse to: detect and resolve conflicts; scope the system and define interfaces with the environment; translate system requirements into software requirements.
- Analysis involves:
 - Classification
 - Conceptual Modelling
 - Architectural Design and Requirements Allocation
 - Requirements Negotiation

SEOC1 – 2003-4

VolBank: Analysis continued

- Classification:
 - Functional: the system *shall* allow a volunteer to be added to the register of volunteers, the following data will be recorded: ...
 - Non-functional:
 - The system *shall* ensure confidentiality of personal data and will not release it to a third party
 - The system shall ensure the safety of all participants
 - Process or product: the systems shall be developed under the relevant ISO 9001 standard.
 - Also priority of feature, scope, volatility

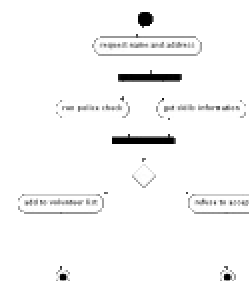
SEOC1 – 2003-4

VolBank: Conceptual Modelling

- Begin to identify classes of object and their associations: volunteer, contact details, match, skills, organisation, needs
- Start to consider some high level model of the overall workflow for the process using modelling tools.

SEOC1 – 2003-4

VolBank



SEOC1 – 2003-4

VolBank: Design and Allocation

- How do we allocate: The system shall ensure the safety of all participants? Further analysis to determine principal threats:
 1. Safety of the volunteer from hazards at the work site.
 2. Safety of the organisations from hazards of poor or inadequate work
 3. Safety of people from volunteers with behavioural problems.
 4. ...
- Design might allow us to allocate 1 to an information sheet, 2 to a rating component and procedures on allocating work, 3 to external police register.

SEOC1 – 2003-4

VolBank: Negotiation

- Safety and Privacy requirements may be inconsistent – need to modify one or both:
 - Privacy: only authorised releases for safety checks will be permitted and there is a procedure for feeding back to the individual if a check fails.
- Some requirements may be achievable but only at great effort – attempt to downscale e.g. it may be too much effort to implement a fault reporting system in the first release of the system

SEOC1 – 2003-4

Other Activities

- Constructing specifications:
 - System requirements definition: customer facing, at system level
 - Software Requirements Specification: developer facing, at software level.
- Requirements validation: key activity aim to get as much as possible – define the acceptance test with stakeholders.
- Requirements Management: requirements change because the environment changes and there is a need to evolve, need tools to manage the collection and maintain *traceability*.

SEOC1 – 2003-4

Summary

- Requirements engineering is a very imprecise activity
- RE is the key activity to constructing good systems cheaply – poor requirements lead to very poor systems.
- Issues are very wide ranging and negotiating agreement between all the stakeholders is hard.
- In some application areas it may be possible to use a more formal notation to capture some aspects of the system (e.g. control systems, compilers, ...)

SEOC1 – 2003-4