A Roadmap’s Roadmap

- **A little motivation**
  » Or, why RE is important

- **A little background**
  » Or, before we begin RE

- **A roadmap**
  » Or, what is RE?

- **“You are here”**
  » Or, on the RE state-of-the-art

- **A little speculation**
  » Or, some open research issues ...
Quality & Requirements

- **Quality** software systems
  - meeting stakeholder needs

- **Primary measure of success**
  - is degree to which a system meets the purpose for which it was intended.

  **Errors cost more** to correct, the longer they go undetected

Requirements Engineering (RE)

- **Requirements** are:
  - expressions of **stakeholder needs** of a **system** to achieve particular **goals**.
  - expressed in the vocabulary of the **problem domain**, rather than the system (solution) domain.

- **Requirements Engineering** is about:
  - Discovering stakeholder goals, needs, and expectations
  - Adjusting stakeholder expectations
  - Communicating these to system implementers.
A Definition of RE

“Requirements engineering is the branch of systems engineering concerned with the real-world goals for, services provided by, and constraints on a large and complex software-intensive system. It is also concerned with the relationship of these factors to precise specifications of system behaviour, and to their evolution over time and across system families.”

[adapted from Zave 1997]

Orientation

- Foundations
- Context and Groundwork
- Eliciting Requirements
- Modelling and Analysing Requirements
- Communicating Requirements
- Agreeing Requirements
- Evolving Requirements
- Integrated Requirements Engineering
Foundations of RE

- Computer Science
- Logic
- Linguistics
- Systems Theory
- Cognitive Psychology
- Anthropology
- Sociology
- Philosophy ... epistemology ... phenomenology ... ontology ...

Context and Groundwork

- **Context**
  - Process improvement and maturity
  - Contract and procurement procedures
  - Organisational setting
  - Personnel and staffing

- **Groundwork**
  - Feasibility
  - Risk

[from Finkelstein 1993]
Eliciting Requirements - what & where

- Requirements elicitation is partly a process of discovering stakeholder expectations, and adjusting these expectations.

- Things to elicit
  - Boundaries
  - Stakeholders
  - Goals
  - Tasks ... use cases ... scenarios
  - Feasibility
  - Risk

- Where to elicit requirements from
  - Stakeholders
  - Application domain
  - Existing documentation

Eliciting Requirements - how

- Traditional techniques
  - Questionnaires, surveys, interviews, analysis of existing documentation, etc.

- Group elicitation techniques
  - Brainstorming, focus groups, RAD/JAD workshops, etc.

- Prototyping
  - For early feedback from stakeholders

- Model-driven techniques
  - Goal-based, scenario-based, etc.

- Cognitive techniques
  - Protocol analysis, card sorting, laddering, etc.

- Contextual techniques
  - Ethnography, conversation analysis, etc.
Modelling and Analysing Requirements

- Enterprise modelling
- Data modelling
- Behavioural modelling
- Domain modelling
- Modelling non-functional requirements (NFRs)

Analysing Requirements Models
- Animation
- Automated reasoning
- Consistency checking

Communicating Requirements

- RE facilitates communication among stakeholders
- Requirements documentation
  - is often the focus of such communication
  - affects choice of specification language
  - sometimes makes use of documentation standards
- Requirements traceability
- Requirements management
Agreeing Requirements

- To design and implement a system, the requirements have to be **agreed**

- To get agreement requirements have to be
  - Validated
  - Negotiated, and conflicts resolved
  - Prioritised

Living with Inconsistency

Evolving Requirements

- **Successful systems will evolve**
  - When the environment in which they operate changes

- **Managing change** is a fundamental RE activity
  - Adding new requirements & requirements scrubbing
  - Fixing errors & managing inconsistency
  - Impact analysis & configuration management

- **Requirements for product families**
  - Need to identify core requirements
  - Reuse of requirements
  - COTS
  - Software architectures
Integrated Requirements Engineering

- **Method engineering**
  - Integrating notations and techniques

- **Problem Frames**
  - Identifying well-understood problems, offers the possibility of selecting corresponding, appropriate, well-understood solutions

- **Tools**
  - DOORS, Requisite Pro, Cradle, RTM, etc.

---

You Are Here!

- Modelling in **context**
- Describing indicative and optative properties of the **environment**

- **Inconsistency happens, live with it!**

- **The RE Community:**
  - REJ, ISRE, ICRE, REFSQ, REP, ...
Journey Planner – a research wish list

- Richer models for capturing and analysing non-functional requirements.
- Techniques for modelling and analysing properties of the environment
  » to deal with incomplete, inconsistent & evolving models
- Reuse of requirements models.
  » to adapt products into product families
- Bridging the gap between elicitation approaches based on contextual enquiry and more formal specification and analysis approaches.
A finer grain process?

Implementation Dependence

Dependent

<table>
<thead>
<tr>
<th>General</th>
<th>Detailed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>Design</td>
</tr>
</tbody>
</table>

Specification

(adapted from Moffett 1999)

Some difficult questions

- What is a requirements engineer?
  - A software architect?
  - A systems engineer
  - ...?

- The end of RE, as we know it?
  - Refinement - not realistic?
  - Documentation - not necessary?
  - Time scales - too long?