

MiRE – Minimum Requirements Engineering Draft

Uolevi Nikula, LUT, Department of
Information Technology, tel. +358 5
621 2839, Uolevi.Nikula@lut.fi

Agenda

-
- * 12.15 – 12.45 Look at MiRE
 - * 12.45 – 13.15 Background to MiRE
 - * 13.15 – 13.45 The MiRE Requirements Document
 - * 13.45 – 14.15 The MiRE Process
 - * 14.15 – 14.30 Break
 - * 14.30 – 14.50 Tool Support for Requirement Engineering
 - * 14.50 – 15.00 MiRE Wrap-Up
 - * 15.00 – 15.30 SFRM, prof. Sajaniemi
 - * 15.30 – 15.45 Discussion and closing

Look at MiRE

- ✦ The MiRE Document
- ✦ The MiRE Process
 - ◆ Techniques Overview 1/2
 - ◆ Techniques Overview 2/2
 - ◆ The MiRE Process Steps
- ✦ Tool Support
- ✦ Look at MiRE Summary

The MiRE Document

- ✦ Development ideas
 - ◆ minimize writing, mainly with tables
 - ◆ placeholder for all normal topics, works as a requirements repository
 - ◆ two options: fast and quality tracks

MiRE Documents

- ✦ Requirements Document
 - ◆ Appendix 1: Glossary
 - ◆ Appendix 2: Typical Computer Configuration
 - ◆ Appendix 3: Use Case Descriptions
 - ◆ Appendix 4: Detailed Requirements
 - ◆ Appendix 5: Data Dictionary (for later use)
 - ◆ Appendix 6: Changes
 - ◆ Appendix 7: Change Requests
- ✦ Interface Specification
- ✦ User Manual
- ✦ Examples
- ✦ Templates walkthrough (MS-Word files)

The MiRE Process

- ✦ Development ideas
 - ◆ invisible – not an end in itself
 - ◆ based on familiar techniques that are used in a systematic way
 - ◆ integrates with the document templates
 - ◆ a suggestion

Techniques Overview 1/2

✧ Introspection

- ◆ Image what the user would do in the given situation

✧ Document reviews

- ◆ Find related documents and study them

✧ Interviews

- ◆ Interview people, either with a prepared list of questions (structured) or without it (unstructured)

✧ Electronic requirements

- ◆ Use emails, web searches etc. to find more information from stakeholders or web

Techniques Overview 2/2

✧ Prototypes

- ◆ Paper mock-ups, user manuals or executable programs that give an idea about the future system's properties. Horizontal ones are shallow but wide (e.g. UI) while vertical ones are deep but limited in their functionality (e.g. testing interface to system xyz).

✧ Workshops

- ◆ A meeting organized to work together on some topic. Typically 5-10 people and a facilitator who runs the workshop to assure focused and timely proceeding.

✧ Reviews/inspections

- ◆ A meeting where material is discussed to find problems in it. The material is distributed beforehand and each participant has studied it alone before the meeting. In the meeting the issues are raised and recorded for later resolving.

The MiRE Process Steps

1. Establish objectives
2. Understand background
3. Organize knowledge
4. Elicit requirements
5. Prepare for problems
6. Prioritize requirements
7. Complete requirements document
8. Analyze the requirements document
9. Validate the requirements document

Iterative Process Model

- ✦ The question with an iterative process model is how many iterations one should do?
 - ✦ A good guideline is four iterations
 - ◆ “We do not want to specify that there always must be four iterations, only that there are four mind-sets to adopt throughout requirements. The iteration names are
 - *Facade* – Outline and high-level descriptions
 - *Filled* – Broadening and deepening
 - *Focused* – Narrowing and pruning
 - *Finished* – Touching up and fine-tuning.”
- Kulak and Guiney 2000, p. 55
- ✦ Process steps walkthrough (MS-Word file)

Tool Support

- ✦ This is an open question but seems important for a ready-to-hand method
- ✦ We will discuss this later today

Look at MiRE Summary

- ✦ The requirements document template and process proposals exist
- ✦ These are initial version and they will change still
- ✦ Tool issue is open
- ✦ Any feedback is appreciated

Background to MiRE

- ✦ MiRE as a Research Effort
- ✦ MiRE as an Industrial Development Effort
- ✦ Requirements Improvement Roadmap
- ✦ New Findings from Literature
- ✦ Background Summary

MiRE as a Research Effort

- ✦ Our research questions are
 1. Can a systematic requirements engineering method be taken in use in small office information system projects with a two person-weeks' effort?
 2. Does a table based minimum requirements document provide real benefits for software development?
 3. Do developers shift to the quality track from the fast track with increasing experience?
 4. Is MiRE a ready-to-hand method?
 5. Is MiRE an invisible method?

MiRE as an Industrial Development Effort

✦ Our goal is to support software development by providing

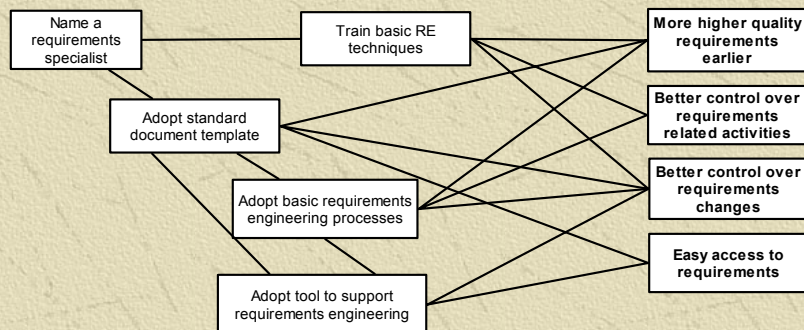
1. easy access to requirements
2. more higher quality requirements earlier in the development cycle
3. better control over requirements related activities
4. better control over requirements changes

April 8, 2002

un

15

Requirements Improvement Roadmap



April 8, 2002

un

16

New Findings from Literature

- ✦ 1/5: Change in Failure Profiles
- ✦ 2/5: A Risk Identification Framework Study
- ✦ 3/5: Related Research Areas
- ✦ 4/5: Agile/Lightweight Methods
- ✦ 5/5: Against Method-ism: exploring the limits of method

1/5: Change in Failure Profiles

Percentage overrun	Cost overrun-%		Time overrun-%	
	Chaos'95	Cutter'01	Chaos'95	Cutter'01
0-50	47	88	22	91
51-100	30	8	20	5
101-200	10	2	35.5	3
>200	13	2	12.3	0

- ✦ Based on Cutter Consortium's study results a clear reduction in project overruns has incurred but it does not reveal reasons for this
- ✦ According to SEI the amount of companies in initial level has dropped from 60.5% to 40.7% while the companies in repeatable and defined levels have increased from 22.5% and 14.7% to 32.4% and 19.3% respectively between 1997 and August 2001

2/5: A Risk Identification Framework Study

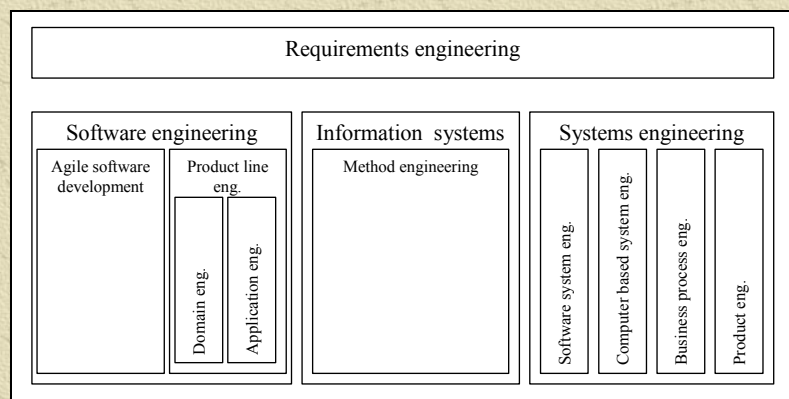
- ✦ [Keil, Cule, Lyytinen, Schmidt 1998]
- ✦ 6 of the identified 11 universal risk factors were somehow requirements related
- ✦ The top three risks
 1. a lack of top management commitment to the project
 2. a failure to gain user commitment
 3. a misunderstanding the requirements

April 8, 2002

un

19

3/5: Related Research Areas



April 8, 2002

un

20

4/5: Agile/Lightweight Methods

- ✦ Agile Software Development Alliance formed in 2001 [agilealliance.org]
 - ◆ E.g. XP, DSDM, Crystal Methodologies, SCRUM, Feature-Driven Development
- ✦ Heavyweight document centric software development approaches
 - ◆ Rational Unified Process
 - ◆ Ones supporting SEI's CMM or ISO 9000
- ✦ Heavyweight methods dominate still but in 2003 50% of the companies will have more than 50% of projects defined as agile [Cutter'01]

5/5: Against Method-ism: exploring the limits of method

- ✦ [Introna and Whitley 1997]
- ✦ A methodology should be ready-to-hand
 - ◆ Otherwise it will break down and be ignored in the pragmatics of getting the job done
- ✦ Tools and methods that are used in order to get the job done need to be invisible to the users
 - ◆ The user must either become seamlessly integrated with the technology (e.g. spectacles) or the technology must be seamlessly merged with the world (e.g. a car speedometer is merged with the road and the car's movement on it)

Background Summary

- ✦ MiRE development tries to address both the academic research goals and the practical software development needs
- ✦ The requirements improvement starts by assigning somebody the role of requirements engineer
- ✦ Literature does provide also interesting research results

The MiRE Requirements Document

- ✦ Start using the template as it is or modify it as you see fit
- ✦ A standard structure is suggested
 - ◆ Remove extraneous topics or tag them “N/A”
 - ◆ When appropriate
 - Get new topics from the comparison table
 - Add your own topics
- ✦ Notice both the fast and the quality track
 - ◆ Fast: record only the issues in the basic template
 - ◆ Quality: use Detailed Requirements Cards (Appendix) to record requirement details
- ✦ See the templates

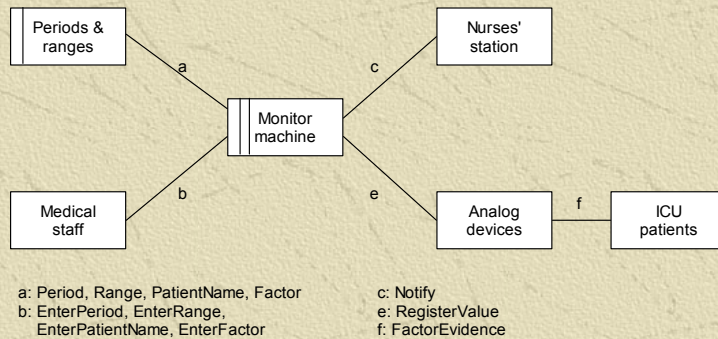
The MiRE Process

- ✦ Example diagrams in the following five slides
- ✦ See the process steps in another file (MS-Word)

Diagrams in MiRE

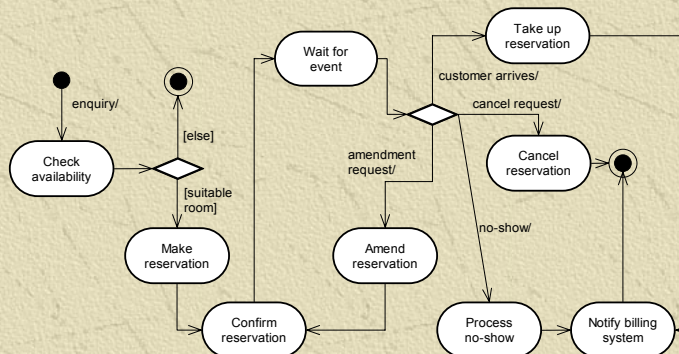
- ✦ Diagrams serve both the analysis and specification purposes by
 - ◆ Making the developer think carefully about the problem
 - ◆ Resulting in a clear figure describing the problem
- ✦ Start with
 - ◆ Context diagram
 - ◆ Dialog map
 - ◆ UML class diagram for data modeling
 - ◆ UML activity diagram for business process modeling
- ✦ We will next take a look at an example of each of these

1/4: Example Context Diagram



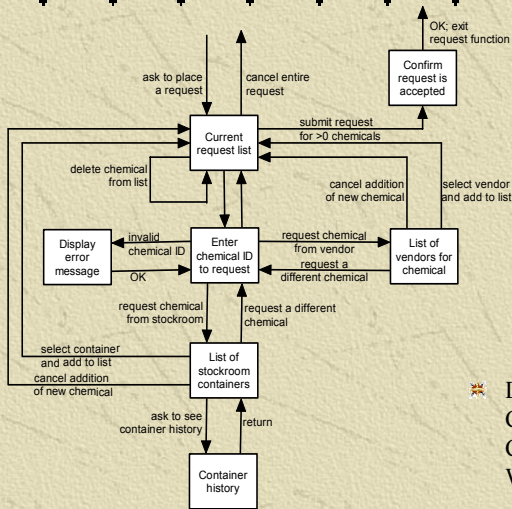
✧ Patient monitoring annotated context diagram, Jackson 2001, p.34

2/4: Example Business Process Model



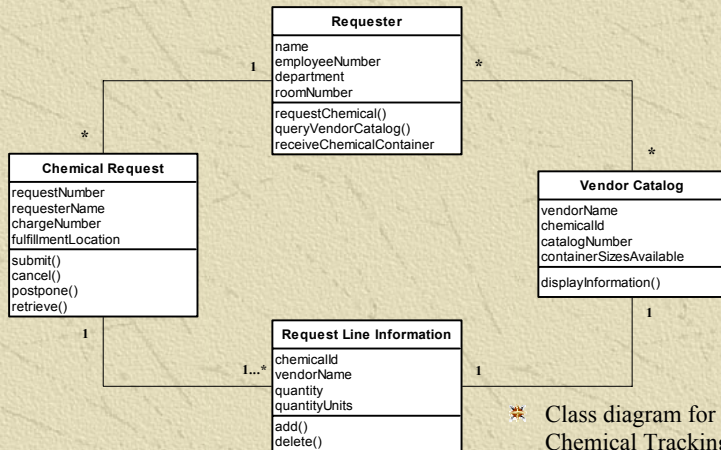
✧ Business process for hotel reservation with UML activity diagram, Cheesman and Daniels 2000, p.68

3/4: Example Dialog Map



✘ Dialog map for the Request a Chemical use case from the Chemical Tracking System, Wiegers 1999, p. 189

4/4: Example Data Model



✘ Class diagram for part of the Chemical Tracking System, Wiegier 1999, p.191

Tool Support for Requirement Engineering

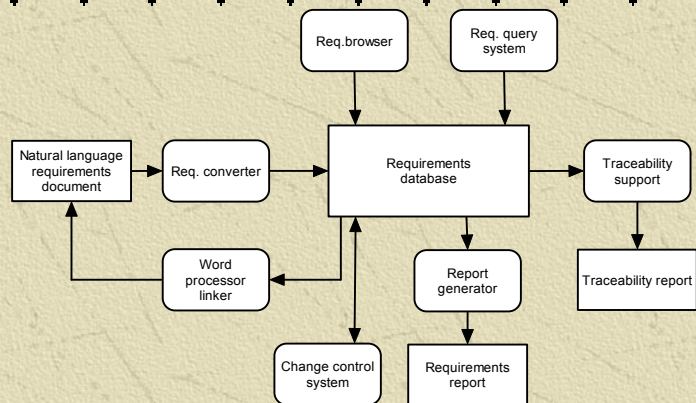
- ✧ A Requirements Management System
- ✧ Different Types of Tools
- ✧ General Questions on Tools
- ✧ Some Tool Suggestions
- ✧ Tool Support for Requirement Engineering Summary

April 8, 2002

un

31

A Requirements Management System



✧ A requirements management system [Kotonya 1998, p.40]

April 8, 2002

un

32

Different Types of Tools

✦ Two types of tools

- ◆ Modeling and validation tools (diagrams, formal models, etc.)
- ◆ Management tools
 - Requirements management tools
 - Change management tools

✦ Requirements management tools

- ◆ Document-centric
- ◆ Database-centric

General Questions on Tools

✦ Operating system

- ◆ Windows
- ◆ Linux
- ◆ Other

✦ Environment for requirements management tool

- ◆ integrated
- ◆ stand-alone

✦ Required output quality (readers)

- ◆ internal development
- ◆ customers and/or auditors

✦ Any internal rules/constraints on tool selection?

✦ Do you find tool support unnecessary – nice – critical?

Some Tool Suggestions

✠ Windows environment

- ◆ MS-Word, MS-Excel etc.
- ◆ SmartDraw (www.smartdraw.com) for drawing
- ◆ sfrm – Saja presentation next

✠ Unix/Linux environment

- ◆ ?

Tool Support for Requirement Engineering Summary

- ✠ Special purpose tools for requirements engineering and especially management exist
 - ◆ INCOSE comparison has 15 tools
 - ◆ Specialized tools are seldom cheap
- ✠ Requirements management tools include often many features
 - ◆ INCOSE questionnaire had c. 65 questions or features
 - ◆ Most of the slots in the comparison table are filled...
- ✠ We will consider the tool support for MiRE closer, too

MiRE Wrap-Up

✧ Next steps

- ✧ This was draft – tuning it is necessary
- ✧ Provide feedback & comments! (Contacts in 1st slide)
- ✧ Writing the thesis
- ✧ Workshop June 6th (or 3rd), 2002
- ✧ Resolving the tool issue

✧ My questions

- ✧ Anyone interested in case studies in fall?
- ✧ Do you have any improvement ideas for MiRE?
- ✧ Do you have any comments on MiRE?

✧ Newsflash

- ✧ Robertson's software and requirement seminar 17.5.2002

References

- ✧ ANSI/IEEE Standard 830-1998 (1998). ANSI/IEEE Standard 830-1998. IEEE Recommended Practice for Software Requirements Specifications. The Institute of Electrical and Electronics Engineers. New York, NY, IEEE Computer Society Press: 32.
- ✧ Cheesman, J. and J. Daniels (2000). UML Components: A Simple Process for Specifying Component-Based Software, Addison-Wesley.
- ✧ Davis, A. M. (1993). Software Requirements: Objects, States, and Functions, Prentice Hall.
- ✧ Davis, A. M., S. Overmyer, et al. (1993). Identifying and Measuring Quality in Software Requirements Specification. First International Metrics Symposium, IEEE Computer Society.
- ✧ Dorfman, M. and R. H. Thayer (1990). Standards, Guidelines, and Examples on System and Software Requirements Engineering. Los Alamitos, CA, IEEE Computer Society Press.
- ✧ Giakoumakis, E. A. and G. Xylomenos (1996). "Evaluation and selection criteria for software requirements specification standards." Software Engineering Journal 11(5): 307-319.

References

- ✦ IEEE/EIA 12207.1-1997 (1997). IEEE/EIA 12207.1, Guide for ISO/IEC 12207, Std for Information Technology - Software life cycle processes - Life cycle data. Industry Impl. of Int Std ISO/IEC 12207 : 1995. New York, NY, IEEE Computer Society Press: 30.
- ✦ Kovitz, B. M. (1998). Practical Software Requirements: A Manual of Content and Style., Manning Publishing Company.
- ✦ Kulak, D. and E. Guiney (2000). Use cases: Requirements In Context. Boston, Addison-Wesley.
- ✦ Lauesen, S. (2002). Software Requirements - Styles and Techniques. Harlow, Addison-Wesley.
- ✦ Leffingwell, D. and D. Widrig (1999). Managing Software Requirements: A Unified Approach. Reading, Massachusetts, Addison-Wesley.
- ✦ Pressman, R. S. (2001). www.rspa.com/docs.
- ✦ Robertson, S. and J. Robertson (2001). Volere Requirements Specification Template, Edition 8. www.systemsguild.com, Atlantic Systems Guild. 2001: 52.
- ✦ STRÍ TS2 (1991). Modelling a Software Quality Handbook. Reykjavik, Iceland, Icelandic Council for Standardization (STRÍ).
- ✦ Wiegers, K. E. (1999). Software Requirements. Redmond, Washington, Microsoft Press, www.processimpact.com.