Software Engineering
Lecture 2 – Requirements

Bogumiła
Hnatkowska

Lecture objectives

- Define key terms used in requirements phase.
- Present basic techniques for requirements elicitation.
- Define elements of software requirements specification.

Requirements – introduction

- Every (mature) model life cycle contains requirements specification stage!
- Requirements specification is the basic input for further stages of software development (analysis, design, coding, testing)
- The requirements specification phase may be preceded by so called feasibility study

Feasibility study

- A feasibility study is a study made before committing to a project
- Its aim is to:
  - find justification for the system existence, e.g. ROI
  - demonstrate that the proposed system is technically feasible.
- A feasibility study typically includes:
  - Outline of requirements
  - Design of a candidate system architecture
  - Estimations of schedule/costs

Feasibility study

- A feasibility study concerns:
  - Clients: Who is this project for?
  - Scope: What are the boundaries of the project?
  - Benefits: What are the benefits? Can they be quantified?
  - Technical feasibility: What are alternative technical solutions?
  - Plan and resources: What are the estimates of staff, time, equipment, etc.?
  - Alternatives and risks: What are the options if the project is not begun?

What is a requirement?

- A **requirement** is a singular documented need of what a particular product or service should be or do
- It is a statement that identifies a necessary attribute, capability, characteristic, or quality of a system in order for it to have value and utility to a user

Characteristics of requirements

- Roles involved in requirements:
  - Business/system analysts
  - Architects
  - Developers (programmers)
  - Testers

Requirements classification

- Functional requirements – define **what** the system must do (the services of the system) for its users
- Non-functional requirements (quality requirements) – define **under what conditions** the system must operate

FURPS model of requirements

- **F**unctionality – functional capabilities, security
- **U**sability – human factors, consistency, documentation
- **R**eliability – frequency/severity of failures, recoverability, accuracy, mean time to failure
- **P**erformance – speed, resource consumption, throughput, response time
- **S**upportability – testability, extensibility, adaptability, maintainability, compatibility, configurability, installability, localizability, portability

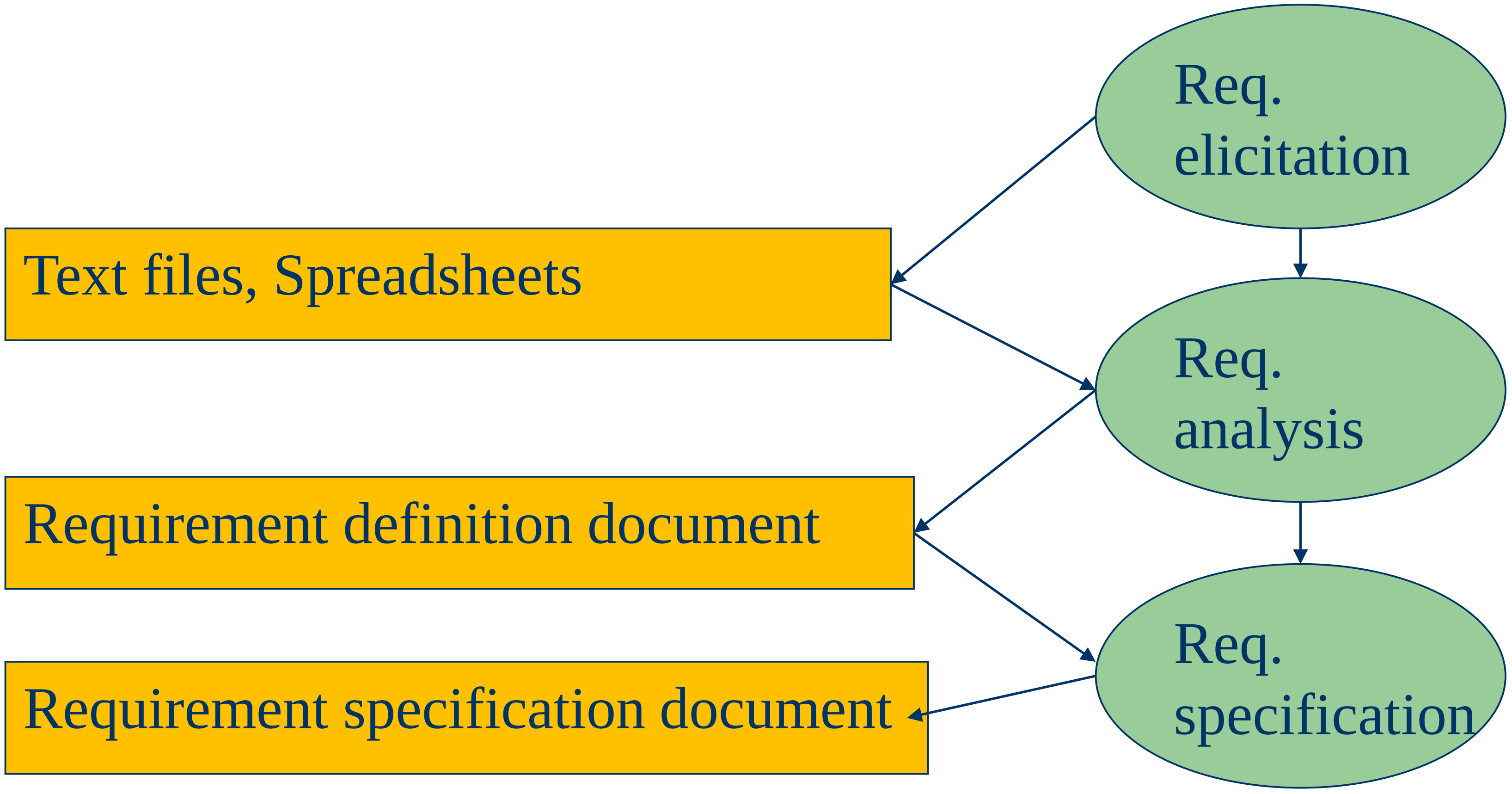
Types of requirements

- Functionality:
  - What will the system do?
  - When will the system do it?
- Data (Functionality/Business rules):
  - What are the input/output data formats?
  - How accurate data must be?
  - To what degree of precision must the calculations be made?
  - Do exist any constraints about the data?
- Users and Human factors (Usability):
  - What kind of training will be required for each type of user?
  - What type of interface should be provided?
- Documentation (Usability):
  - How much documentation is required?
  - What format of documentation is acceptable?
  - To what audience is each type of documentation addressed?

Types of requirements

- Performance demands (Performance)
  - Are there constraints on execution speed, response time, throughput?
- Security:
  - Must access to the system or to information be controlled?
  - How will one user’s data be isolated from others?
  - How often will the system be backed up? Where the back-up copies should be stored?
- Interfaces with other systems (Other constraints):
  - Is the input/output coming from/going to other systems?
  - Is there a prescribed way in which data must be formatted?
- Physical environment (Other constraints):
  - Where is the equipment to function?
  - Are there any environmental restrictions, such as temperature, humidity, magnetic interference?

Requirements specification



Two kinds of requirements documents

- Requirement definition document – written usually in natural language that the customer and developer can understand
- Requirement specification document – rewrites the requirement definition document in technical terms appropriate for the development
- Sometimes a single requirements document servers both purposes

Problems with requirements

- Are not always obvious.
- Come from many sources.
- May not always be easy to express clearly in words.
- Relate to one another.
- Change !!!

Requirements elicitation

- Requirements elicitation (requirements gathering) - the practice of obtaining the requirements of a system from users, customers and other stakeholders.
- Requirements elicitation techniques (examples):
  - Reading existing documents, regulations, etc.
  - Brainstorming
  - Interviews
  - Questionnaires
  - Observations

Requirements analysis

- Requirements analysis encompasses activities that aim in checking gathered requirements in order to eliminate redundancy, ambiguity, inconsistency etc.
- Results are presented as software requirement definition document

What requirements definition document should be like?

- Cohesive
- Complete
- Consistent
- Correct
- Unambiguous
- Feasible (realistic)
- Verifiable
- Traceable

Requirements Definition Document

- Structure:
  - General purpose of the system
  - List of requirements
- Recommendations:
  - Each clause should contain only one requirement
  - Collect similar requirements together, e.g. functional and nonfunctional
  - Organize requirements per user type (role)

Requirements examples

Requirements for Loan Arranger system

- Functional requirements at general level:
  - The system **must** allow the loan analyst to create, view, edit, or delete a loan from a portfolio
  - The system **should** allow the loan analyst to create a new lender
  - The system **can** allow the loan analyst to delete a lender only if there are no loans in the portfolio associated with the lender
- Business rules and guidelines:
  - A single borrower may have more than one loan
  - Each loan must have at least one borrower
  - There are two types of loans based on the amount of the load: regular and jumbo. A regular loan is for any amount less than or equal to $275,000. A jumbo loan is for any amount over $275,000

Requirements examples

- Non-functional requirements:
  - After updating displayed information, the information is refreshed within 5 seconds
  - The system must be available for use by a loan analyst during 97% of the business day
- Constraints:
  - The system should work on a UNIX platform

Exercise

- Propose some functional requirements for Internet Shop app
- Propose some business rules (constraints) for Internet Shop app
- Propose some non-functional requirements for Internet Shop app

How to express requirements?

- Requirements Definition Document:
  - Natural language
  - Decision tables, decision trees (for rule base systems)
- Requirements Specification Document:
  - Specification languages, e.g. UML (see next lecture)

Decision table

- Good for presenting regulations, business rules, classification knowledge
- Consists of 3 parts:
  - Condition rows (complete set in a given area)
  - Action rows
  - Rules (columns)

Decision tables

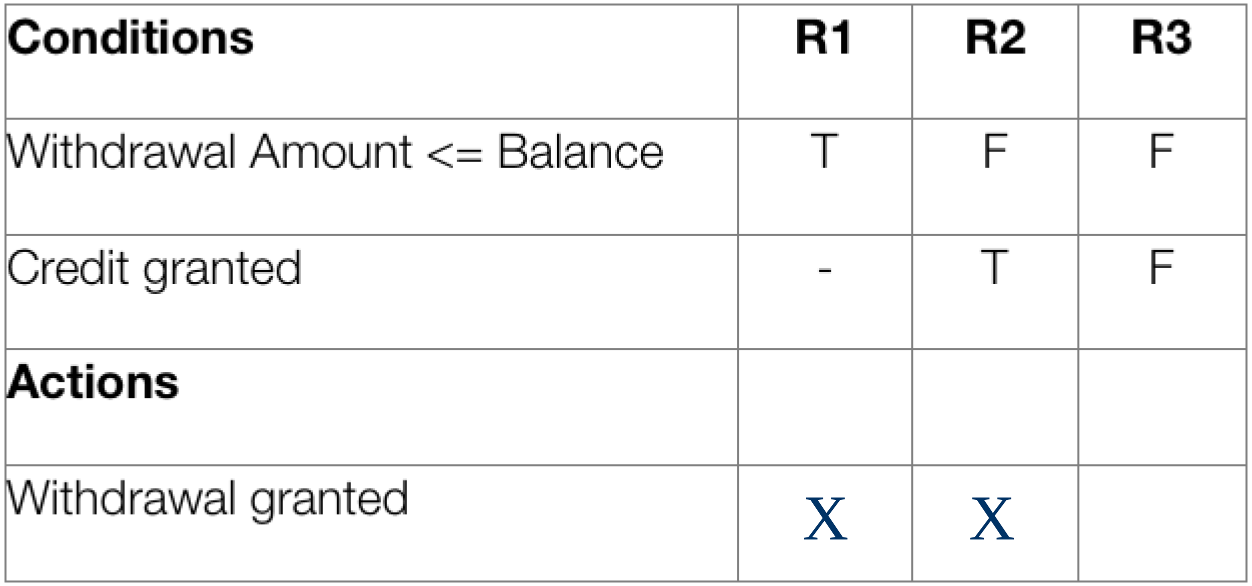
Test definition

	Rule 1	Rule 2	Rule 3
Condition 1	T	T	F
Condition 2	F	T	T
…	_	_	T
Action 1	X		
Action 2		X	
Action 3			X

T – condition must be true
F – condition must be false
_ – condition value is not important (has no influence on the result)
X – action is performed

Decision table example

- Limited format: condition entries are limited to values Yes/No, True/False, etc.; X represents the action to take



Conditions	**R1**	**R2**	**R3**
Withdrawal Amount <= Balance	T	F	F
Credit granted	-	T	F
Actions			
Withdrawal granted	X	X	

Decision table example

- Eextended format: The conditions are written as open ended questions, e.g. Driver Age, and the values as parts of rules, e.g. <21, >=21. Actions can be written directly in columns

Insurance policy	R1	R2	R3	R4
Driver age	<21		>=21	
Engine capacity	<1000	>=1000	<1000	>=1000
Value paid	500	700	400	600

Decision table example

- Mixed format: a combination of limited and extended entries.

Insurance policy	R1	R2	R3	R4
Driver age	<21		>=21	
Engine capacity	<1000	>=1000	<1000	>=1000
Pay 500	x			
Pay 600			x	
Pay 400		x		
Pay 600				x

Decision table – how to prepare

1. Identify conditions and their values
2. Identify actions
3. Compute maximal number of rules
4. Define rules
5. Define actions for each rule
6. Simplify the table (if its possible)

Decision table – exercise

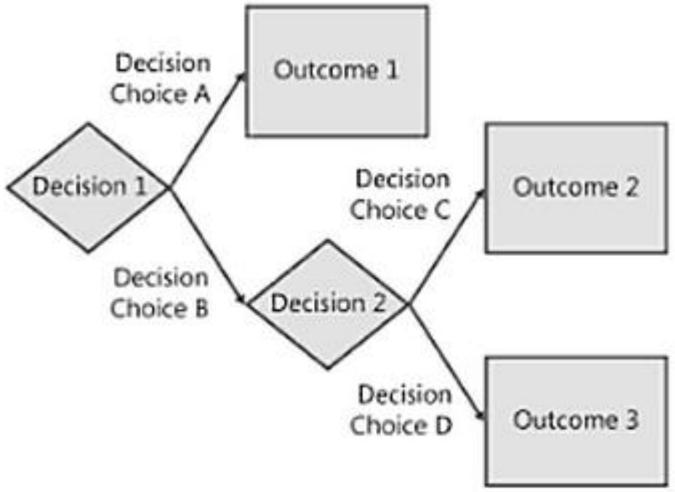
- Propose a decision table (both limitted format, and extended format) that „produces” one of words: „Miss”, „Mrs.”, „Mr.” on the basis of sex and marriage status

Decision table - exercise

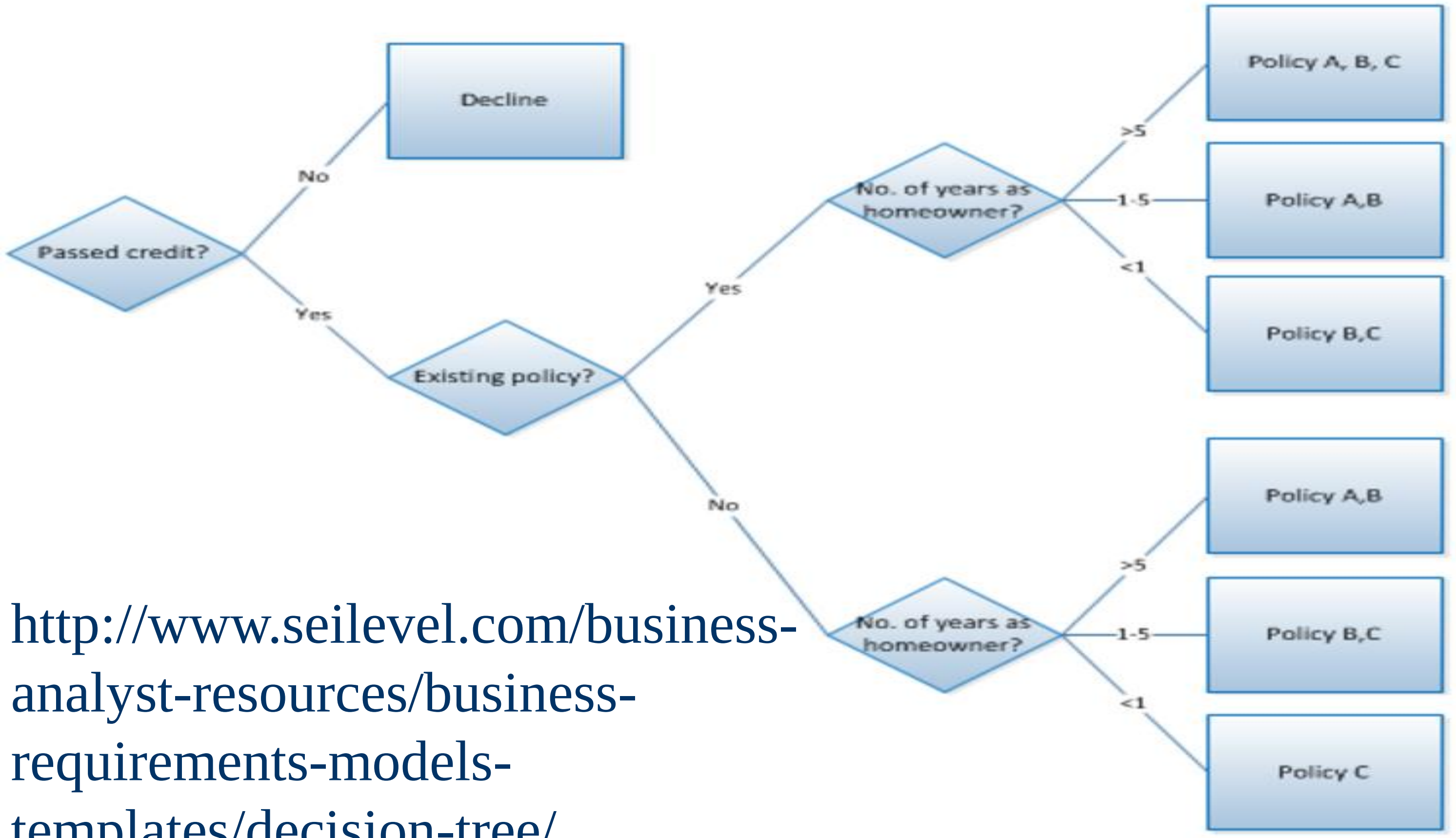
- Define a decision table for the ATM assuming that a client wants to get some money; the transaction takes two inputs: PIN, amount
- The transaction to be successful requires:
  - PIN must be correct, otherwise msg: Incorrect PIN
  - Balance must be greater or equal amount, otherwise msg: Insufficient balance

Decision tree

- Allows to model a complex logic
- Represents a series of decisions
- Elements:
  - Decisions
  - Outcomes
  - Decision choices



Decision tree example



http://www.seilevel.com/business-analyst-resources/business-requirements-models-templates/decision-tree/

Decision tree - exercise

- Translate previously prepared decision table into a decision tree

Summary

- There are two kind of requirements: functional and non-functional. Functional requirements explain what the system should do, and the non-functional ones constrain the behavior in terms of safety, reliability etc.
- The requirements definition and specification documents should describe the problem, leaving solution selection to the designers
- The requirements document should be checked for completeness, correctness, consistency, realism, and more.
- There are many different types of definition and specification techniques. The most popular use the natural language. The other incorporate particular modelling languages like UML or SysML

Revision

- What is a requirement?
- How requirements are classified?
- What activities are performed within requirement stage?
- What two main documents are elaborated within requirement stage?
- What techniques are used for expressing requirements?
- What the requirement specification should be like?

Revision

- Try to find errors in the requirements for calculator software presented below:
  - Users and human factors:
    1. The program will be used by any ordinary computer user without any specific knowledge for simple calculations
    2. No training is needed to use the program.
  - Data:
    3. The input/output numbers can be provided in decimal, hex and binary systems
    4. The program should calculate results with 4 position precision
    5. The program should operate on the numbers that consists of at least 10 numbers (positions)
  - Functionality:
    6. The program should calculate sum, product, and quotient of two numbers given with 2 places precision
    7. The program allows to remember intermediate values in the calculator memory
    8. The program allows to clear the memory
    9. The program allows to use a value from the memory as an argument for operation

    The program must accept input data provided as typical expressions, e.g. 2+3-5
  - Security:
    10. All program operations should be recorded.
    11. Access to the program should be controlled by login and password.