CS445 / SE463 / ECE 451 / CS645

Software requirements specification & analysis

10. The Software Requirements Specification (SRS)

Fall 2010 — Mike Godfrey

Overview

- Software requirements specifications (SRS)
  - IEEE standard for organizing an SRS

- More reading:
  - IEEE Recommended Practice for SRSs, 1998 (available from an on-campus machine via the course web page)

Lecture includes some excerpts from

"Requirements document for an automated teller machine network"

ANY OF:
http://www.cs.toronto.edu/~sme/CSC340F/2005/assignments/inspections/atm.pdf
https://www.academia.edu/27595821/Requirements_document_for_an_automated_teller_machine_network
https://tejalal.files.wordpress.com/2015/09/atm-srs.pdf
https://docplayer.net/925252-Requirements-document-for-an-automated-teller-machine-network.html

[links checked July 2020]

SRSs

- The IEEE Recommended Practice for Software Requirements Specifications (RPSRS):
  - Describes what information should go in an SRS
  - How the information should be arranged
  - Provides several sample outlines for an SRS

SRS contents

- The main issues that the SRS should address are:
  - Functionality
    - What the software is supposed to do
  - External interfaces
    - How the software interacts with people, the system's hardware, other hardware, other software
  - Performance
    - Required speed, availability, response time, recovery time of various software functions

SRS contents

- Quality attributes (NFRs)
  - What are the portability, correctness, maintainability, security, etc. considerations
- Design constraints
  - Design decisions that constrain the set of acceptable solutions: standards, implementation language, policies for data integrity, resource limits, operating environment(s)

SRS contents

- Typically, the SRS does *not* address:
  - Process requirements
  - Design decisions

IEEE SRS organization

Table of Contents

Table of Figures

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, acronyms, abbreviations

1.4 References

1.5 Overview

2. Overall description

2.1 Product perspective

2.2 Product functions

2.3 User characteristics

2.4 Constraints

2.5 Assumptions and dependencies

3. Specific requirements

/ variable organization */*

Appendices

Index

Section 1. Introduction

- More of an introduction to the document than the actual system to be built

1.1 Purpose

- Purpose of the SRS
- Who is the intended audience of this document?
- How it is to be used?
  - e.g., contract between vendor and customer?
- .25-.5 pages

ATM example: Purpose

This document describes the software requirements and specification for an automated teller machine (ATM) network. The document is intended for the customer and the developer (designers, testers, maintainers).

The reader is assumed to have basic knowledge of banking accounts and account services. Knowledge and understanding of UML diagrams is also required.

Section 1. Introduction

1.2 Scope

- Name of the software product
- Overview of the product – what it will / will not do
- Summary of the application of the software, including benefits, goals
- The boundaries of the product
- .25-.5 pages

ATM example: Scope

The software supports a computerized banking network called YouBank. The network enables customers to complete simple bank account services via automated teller machines (ATMs) that may be located off premise and that need not be owned and operated by the customer's bank. The ATM identifies a customer by a cash card and password. It collects information about a simple account transaction (e.g., deposit, withdrawal, transfer, bill payment), communicates the transaction information to the customer's bank, and dispenses cash to the customer. The banks provide their own software for their own computers. The YouBank software requires appropriate record keeping and security provisions. The software must handle concurrent accesses to the same account correctly.

Section 1. Introduction

1.3 Acronyms, Abbreviations, Definitions, Notational Conventions

- Usually for domain-level definitions used in the SRS
  - Project-related definitions should be in the Glossary.
  - Could just throw all defs into the Glossary
- Explain any naming conventions you develop to help you write the document
- Explain any notational conventions for any deviations from standard UML notation
  - For example, you can be creative with fonts or colour to denote different types of names, e.g., red for attributes, blue for operations.
  - Colored information stands out in a state diagram in which transitions are labeled with events, conditions, activities, etc.

ATM: Definition vs. Abbreviation

- Definition:
  - *Account* – *A single account at a bank against which transactions can be applied. Accounts may be of various types with at least checking and savings. A customer can hold more than one account.*

- Abbreviation:
  - *maxDailyWD* – *The maximum amount of cash that a customer can withdraw from an account in a day (from 00:00 AM to 23:59 PM) via ATMs.*

ATM example

- Definitions:
  - ATM
  - Bank
  - Bank computer
  - Cash Card
  - Customer Transaction
- Abbreviations (constants)
  - maximum withdrawal per day and account
  - maximum withdrawal per transaction
  - minimum withdrawal per transaction
  - minimum cash in the ATM to permit a transaction
  - total funds in the ATM at the start of a day

Section 1. Introduction

1.4 References

- Your sources of information, such as
  - Pre-existing project documentation
  - Documentation of stakeholder interviews

    e.g., meeting minutes, videos, email
  - External info sources

    e.g., a textbook on telephony, web pages

Section 1. Introduction

1.5 Overview

- Brief description of the structure of the rest of the SRS, especially:
  - Chosen organization for section 3 (more later)
  - Any deviations from the standard SRS format

ATM: Overview

The rest of this document is organized as follows. Section 2 contains a general description of the ATM network software requirements. Section 3 identifies the specific requirements, including external interfaces, use cases, functional requirements, and behavioral requirements. The document concludes with an appendix of glossary terms.

Appendices consist also of minutes of customer interviews and meetings, and do not constitute additional requirements of the software; all requirements arising from these minutes have been incorporated into the specific requirements in Section 3.

Section 2: Overall description

- This section gives an overall description of the system under development, including general factors that affect the product and its requirements.
  - Do not state specific requirements here; instead, provide a background for those requirements, which are defined in detail in Section 3, and makes them easier to understand.

2.1 Product Perspective

2.2 Product Functions

2.3 User Characteristics

2.4 General Constraints

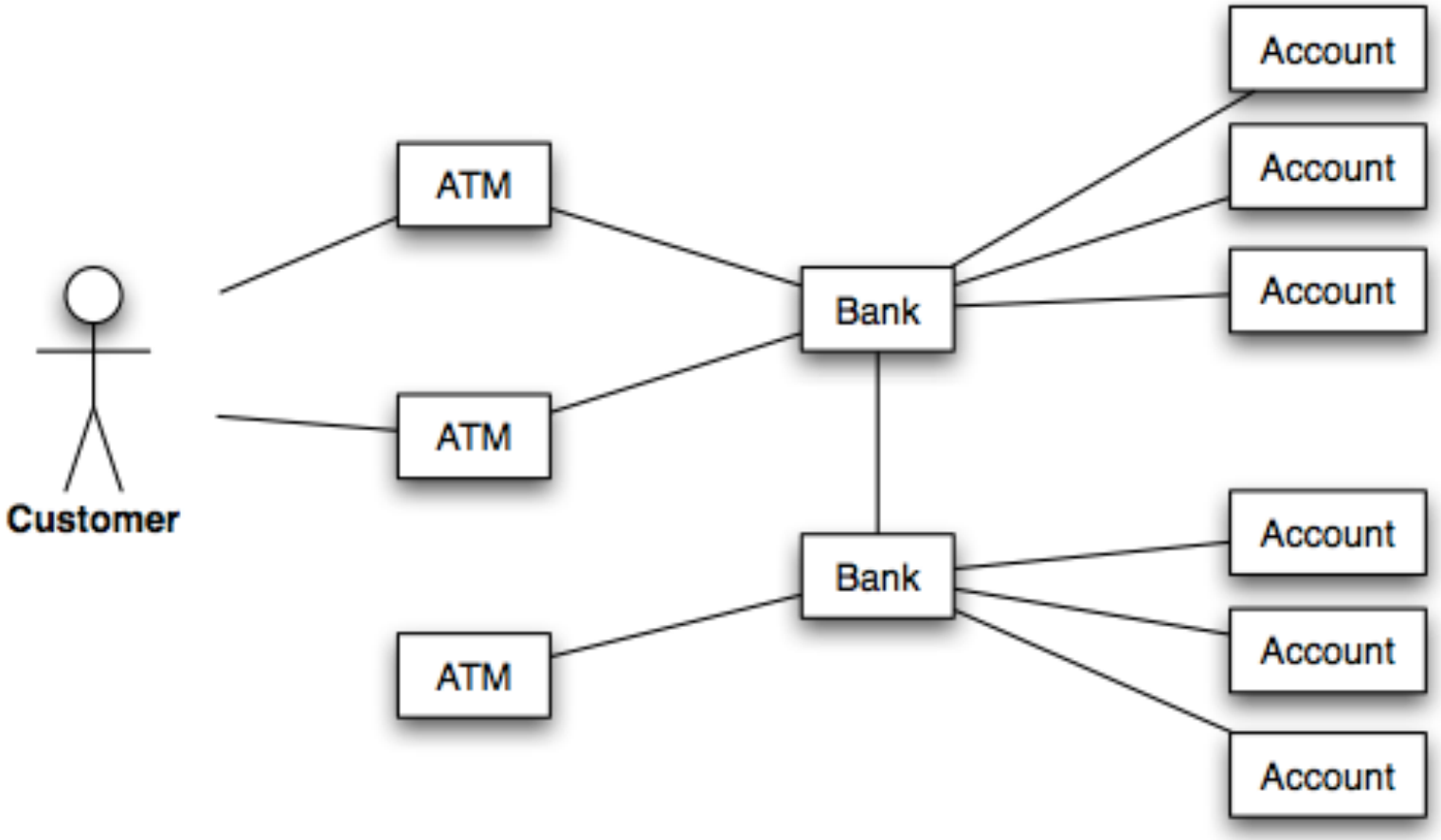
2.5 Assumptions and Dependencies

Section 2: Overall description

2.1 Product Perspective

- Describe the environment of the system
  - e.g., hardware/software components that interact with the system
- Include a context diagram!

2.1 Product perspective



2.1 Product perspective

- A detailed description is not necessary, since interface specifications appear later in the document.
  - Give just an overview of the interfaces to other components in the environment.

- This section includes requirements of the user interface, such a testable usability requirements.
  - This is distinct from the user interface “design” that is described in Section 3.

ATM: Product perspective

The ATM network does not work independently. It works together with the banks' computers and the software run by the network's banks.

Communication interface – The ATMs communicate with the banking systems via a communication network; the protocol used is specified in [Ref1].

Software interface – The messages sent via the communication network are specific to the target banking software systems. At present, two known banking systems will participate in the ATM network. The interfaces to these systems are specified in [Ref2], and [Ref3].

Hardware interface – The software will run on an ATM computer yet to be chosen.

ATM: Product perspective

- *User interfaces*
  - *Customer – The customer user interface should be intuitive, such that 99.9% of all new ATM users are able to complete their banking transactions without any assistance.*
  - *Bank Security Personnel – Bank security personnel are responsible for removing deposits and adding cash to ATMs. There should be a simple interface (e.g., a switch or button) that they can use to initialize the ATM whenever they restock.*
  - *Maintainer – The maintainer is responsible for adding new ATMs to the network and servicing existing ATMs. A maintainer should be possible to add a new ATM to the network within 1 hour.*

Section 2: Overall description

2.2 Product Features

- An overview of the system's main features
  - Need give only a textual list of UC names (or "brief" UC summaries)
  - List should be complete
  - Features will be specified in detail in Section 3

Section 2: Overall description

2.3 User Characteristics

- Document any assumptions you make about the user and any assumptions you make about the background or how much training the user will need to use the system.
  - For example, you could build different user interfaces for knowledgeable and novice users.
- Consider only user characteristics that affect the software requirements

ATM: User characteristics

- *There are several users of the ATM network:*
  - *Customers are simply members of the general public with no special training.*
  - *Bank security personnel need have no special education or experience.*
  - *Maintainers must be experienced network administrators, to be able to connect new ATMs to the network.*

Section 2: Overall description

2.4 General Constraints

- Sources of other constraints on requirements
  - regulatory policies
  - hardware limitations
  - parallel operation
  - audit functions
  - control functions
  - criticality of the application
  - safety and security considerations
  - standards
  - laws

Section 2: Overall description

- Sections 2.1 through 2.3 describe sources of possible constraints on requirements:
  - 2.1 describes existing environment components that might constrain requirements
  - 2.2 describes desired functionality
  - 2.3 describes users' backgrounds that might affect usability issues.
  - You need to consider whether there are any sources of constraints on requirements or design.

- Note that 2.4 isn't NFRs *per se*
  - … but it is a set of sources of possible NFRs

Section 2: Overall description

2.5 Assumptions and Dependencies

- Assumptions about input/environmental behavior, such as
  - hardware never fails
  - ATM casing is impenetrable
  - limited number of transactions per day (sufficient paper for receipts)
  - limited amount of money withdrawn per day (sufficient money)
  - people will naturally avoid railway crossing when gate is down
- What conditions could cause the system to fail?
- What changes in the environment, could cause changes to the software requirements?

Section 3: Specific requirements

- This section of the SRS should contain all of the software requirements and specification.
  - This is the “meat” of the SRS; all UML and UI diagrams go in here! Expected input/output behaviour is detailed here.

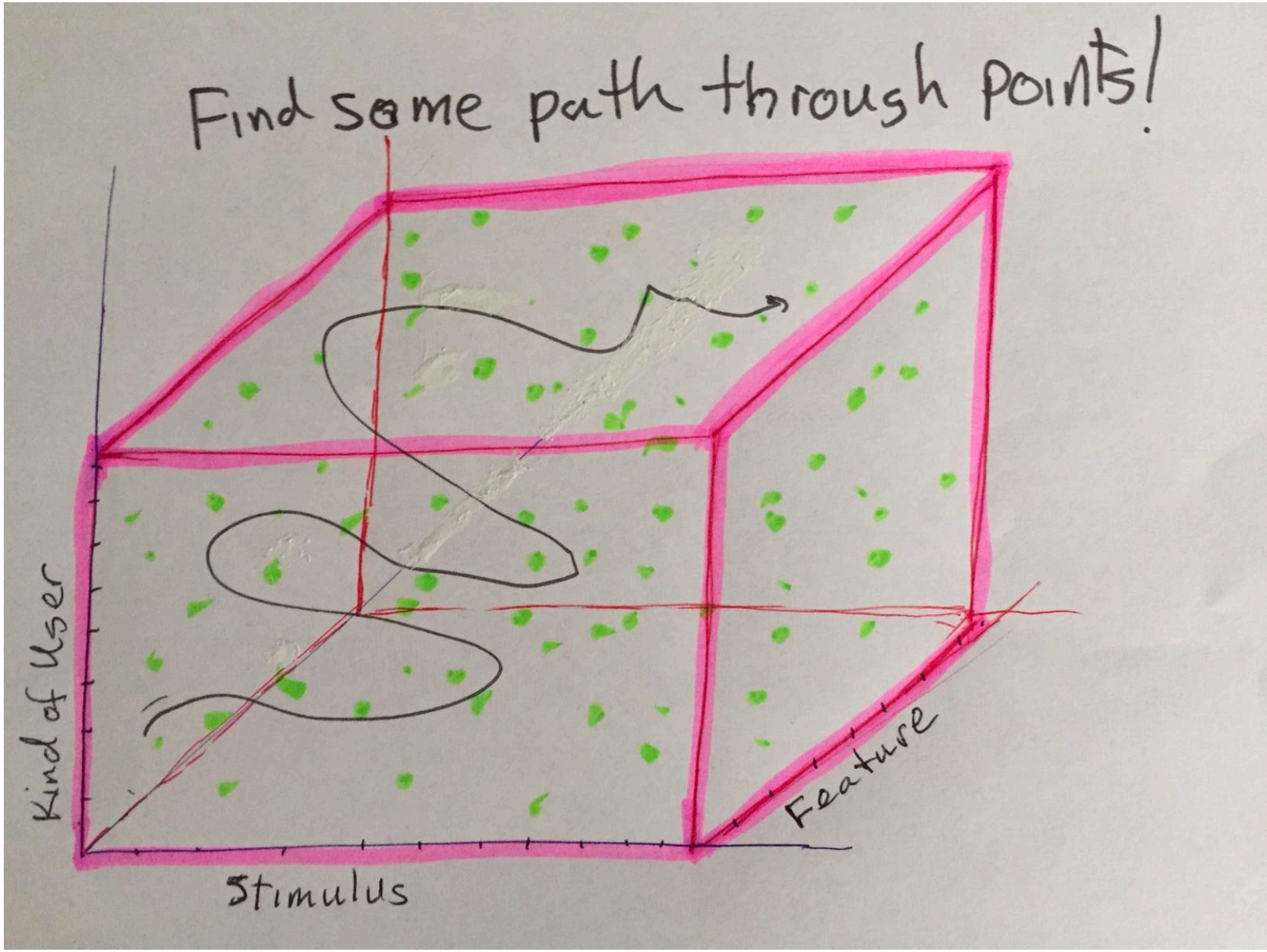
- At a minimum, it should include descriptions of
  - All interfaces to the system
    - Every input (stimulus) into the system
    - Every output (response) from the system
  - All functions performed by the system
    - validity checks on inputs
    - relationship of outputs to inputs
    - responses to abnormal situations (e.g., overflow, error handling)

Section 3: Specific requirements

- Input and output definitions should be consistent among UCs, functional specs, state machines, and UIs.

- Should be at a level of detail sufficient to enable:
  - designers to design a system that satisfies the specification
  - testers to test that the system conforms to the specification

- There are different ways of organizing Section 3
  - We'll look at two approaches briefly


Find some path through points!
Kind of User
Stimulus
Feature

Section 3: IEEE organization

3.1 External Interfaces

- Detailed descriptions of all inputs and outputs
  - Name of input (or output)
  - Description of purpose
  - Source of input or destination of output
  - Valid range, accuracy, and/or tolerance
  - Units of measure
  - Timing Relationships to other inputs/outputs
  - Screen formats/organization
  - Window formats/organization
  - Data formats
  - Command formats

Section 3: IEEE organization

3.2 Functional Requirements

- Use case descriptions
- Sequence diagrams
- Domain model
- Functional specifications
- State machine model
- Constraints

Section 3: IEEE organization

- Can organize functional requirements (Section 3.2) in several ways:
  - Kind of user
  - Features
  - “Stimulus” [use case view]

A.3 Template of SRS Section 3 organized by user class

3. Specific requirements
- 3.1 External interface requirements
  - 3.1.1 User interfaces
  - 3.1.2 Hardware interfaces
  - 3.1.3 Software interfaces
  - 3.1.4 Communications interfaces
- 3.2 Functional requirements
  - 3.2.1 User class 1
    - 3.2.1.1 Functional requirement 1.1
    - .
    - .
    - .
    - 3.2.1.*n* Functional requirement 1.*n*
  - 3.2.2 User class 2
  - .
  - .
  - .
  - 3.2.*m* User class *m*
    - 3.2.*m*.1 Functional requirement *m*.1
    - .
    - .
    - .
    - 3.2.*m*.*n* Functional requirement *m*.*n*
- 3.3 Performance requirements
- 3.4 Design constraints
- 3.5 Software system attributes
- 3.6 Other requirements

IEEE 830-1998 Recommended Practice for Software Requirements Specification

A.5 Template of SRS Section 3 organized by feature

3. Specific requirements
- 3.1 External interface requirements
  - 3.1.1 User interfaces
  - 3.1.2 Hardware interfaces
  - 3.1.3 Software interfaces
  - 3.1.4 Communications interfaces
- 3.2 System features
  - 3.2.1 System Feature 1
    - 3.2.1.1 Introduction/Purpose of feature
    - 3.2.1.2 Stimulus/Response sequence
    - 3.2.1.3 Associated functional requirements
      - 3.2.1.3.1 Functional requirement 1
      - .
      - .
      - .
      - 3.2.1.3.*n* Functional requirement *n*
  - 3.2.2 System feature 2
  - .
  - .
  - .
  - 3.2.*m* System feature *m*
  - .
  - .
  - .
- 3.3 Performance requirements
- 3.4 Design constraints
- 3.5 Software system attributes
- 3.6 Other requirements

IEEE 830-1998 Recommended Practice for Software Requirements Specification

A.6 Template of SRS Section 3 organized by stimulus

3. Specific requirements
- 3.1 External interface requirements
  - 3.1.1 User interfaces
  - 3.1.2 Hardware interfaces
  - 3.1.3 Software interfaces
  - 3.1.4 Communications interfaces
- 3.2 Functional requirements
  - 3.2.1 Stimulus 1
    - 3.2.1.1 Functional requirement 1.1
    - .
    - .
    - .
    - 3.2.1.*n* Functional requirement 1.*n*
  - 3.2.2 Stimulus 2
  - .
  - .
  - .
  - 3.2.*m* Stimulus m
    - 3.2.*m*.1 Functional requirement *m*.1
    - .
    - .
    - .
    - 3.2.*m*.*n* Functional requirement *m*.*n*
- 3.3 Performance requirements
- 3.4 Design constraints
- 3.5 Software system attributes
- 3.6 Other requirements

IEEE 830-1998 Recommended Practice for Software Requirements Specification

Section 3: IEEE organization

3.3 Performance requirements

- Stated in concrete terms, such as:
  - number of terminals to be supported
  - number of simultaneous users to be supported
  - amount and type of information to be handled
  - number of transactions to be processed within a set time period
  - normal workload conditions peak workload conditions

Section 3: IEEE organization

3.4 Design Constraints

3.5 Quality Attributes

- Nonfunctional properties (besides performance), expressed as testable constraints

Dial Tone Deadline		**ID:** NF17	**Importance:** E
Overview: When a caller picks up a phone's headset, the system should issue a dial tone within the specified deadline.			
Fit Criteria:	**Outstanding**	**Target**	**Mininum**
	deadline – 0.1 sec	deadline – 1 s	deadline – 2 s
References: Meeting #5, R9			

CS445 / SE463 / ECE 451 / CS645

Software requirements specification & analysis

10. The Software Requirements Specification (SRS)

Fall 2010 — Mike Godfrey