

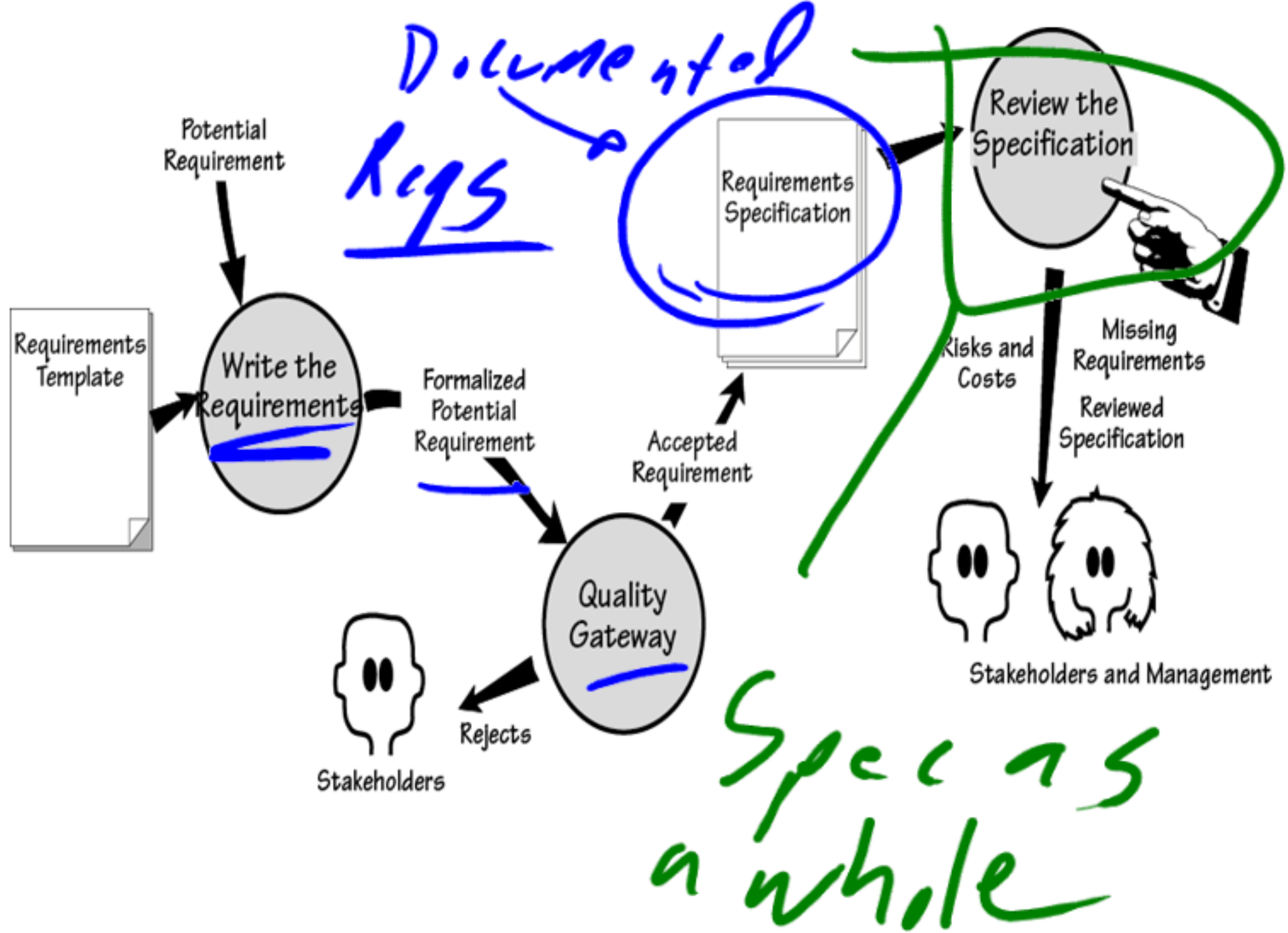


Requirements Completeness

Lecture Objectives:

- 1) Explain why reviewing the requirements specification tends to be an iterative process
- 2) Explain, at a high level, the concept of a Fagan Inspection
- 3) Show how business events can be checked for completeness.
- 4) Explain the meaning of a CRUD check and how it can be used to check requirements scenarios.
- 5) Explain when requirements should be prioritized and list prioritization criteria.
- 6) List categories of risk to a software project
- 7) Explain how risk can be quantitatively assessed for a project.

Reviewing the Requirements



Reviewing the Specification

Iterative Process? ~~+~~ ~~+~~

- Why?

1. Make sure everything is caught.

2. Finding defects may inject new defects into the system

- Should keep a history of the errors uncovered
 - Why?

Worse than the problem. 1/7 of Fixes are

Reviewing the Specification

- Iterative Process?
 - Why?

- ★ • Should keep a history of the errors uncovered ★
 - Why?

To avoid / or aid
in preventing future
defects

Fagan Inspection

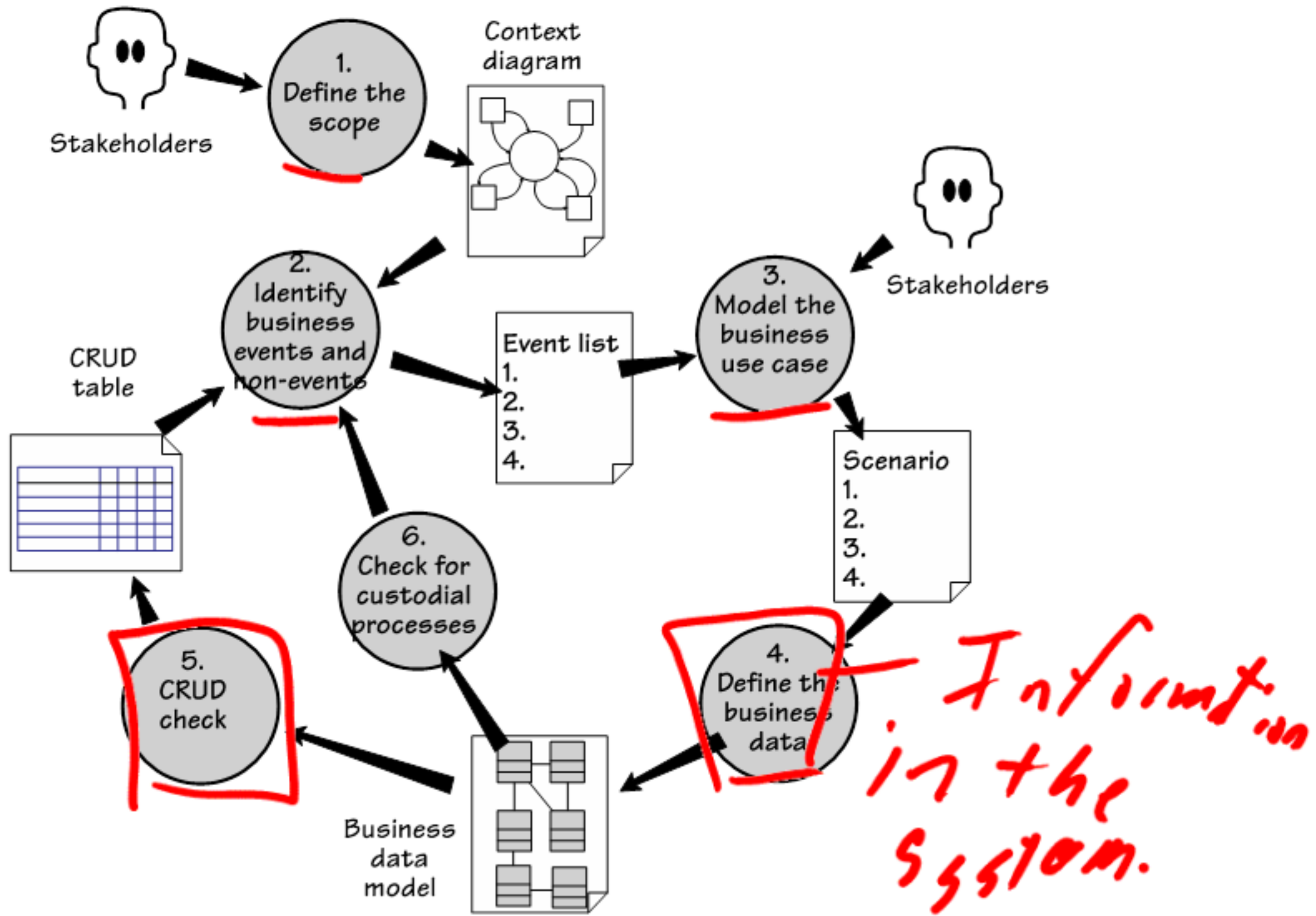
- Formal Process for inspecting software artifacts
 - Discuss more in ~~SE4931~~ *SE3800 next quarter*
- Involves
 - assigning a moderator —
 - Building a checklist of common mistakes —
 - Giving inspectors time to individually inspect the artifact ⇒ *"prepare"*
 - Hold an inspection meeting to discuss errors
 - Limit inspection activities to 2 hours per day
 - Have between 3 and 8 inspectors

1970 started

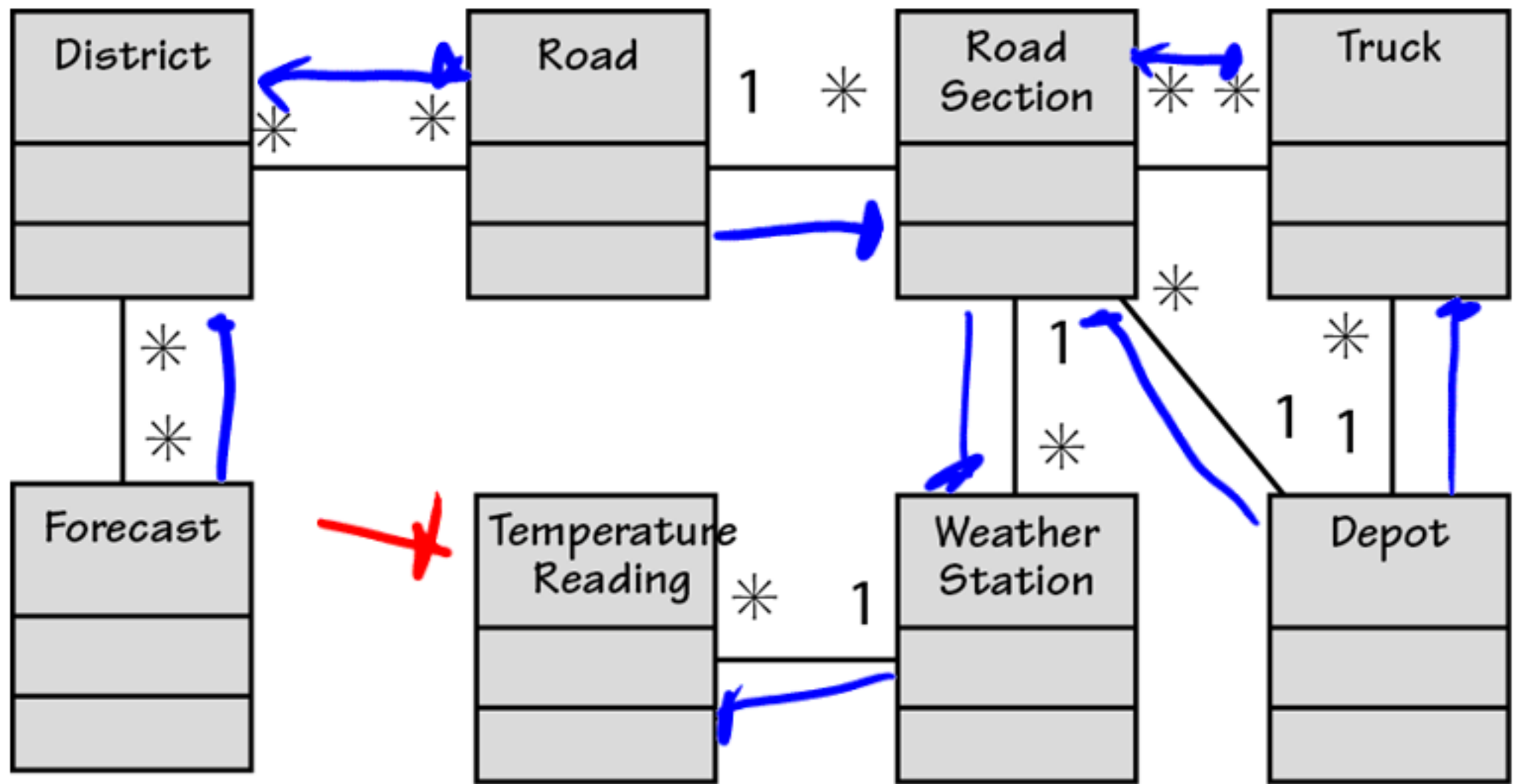
≈ 2hrs

Analyzing the business events

events



Defining the Business Data



High Level Data Model.
⇒ Firstcut of Relations.

CRUD Check

- CRUD
 - Create
 - Reference
 - Update
 - Delete
- A mechanism for checking that all classes within the system are used properly.



CRUD

The CRUD Table

Class	Create	Reference	Update	Delete
Depot				
District				
Forecast	3	Forecast roads		clean up
Road				
Road Section				
Temperature Reading	obtain temps	Monitor Road Conditions	X	?
Truck				
Weather Station				

Table Allows to check the data in our system.

Prioritizing Requirements

- When should you start to prioritize requirements?

~~After done~~
~~When reqs are done~~

As soon as possible
in the project.

- What should you use to prioritize requirements?

User Satisfaction / Dissatisfaction
Dependencies
More important, (Gut feel)

Prioritizing Requirements

- When should you start to prioritize requirements?
- What should you use to prioritize requirements?

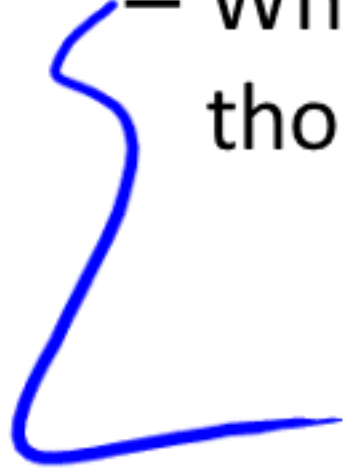
Cost
Value

Time needed
Ease of technological implementation
Benefit
Obligation to obey the law



Prioritizing Requirements

- Discussion:
 - Which requirements do you start with, those that are easy or those that are hard?



"V
ICS"

Prioritization Spreadsheet

Weight Implementation Cost

Volere Prioritisation Spreadsheet
Copyright © The Atlantic Systems Guild 2006

Requirement/Product Use Case/Feature	Number	Factor - score out of 10	%Weight applied	Factor - score out of 10	%Weight applied	Factor - score out of 10	%Weight applied	Factor - score out of 10	%Weight applied	Priority Rating	Total Weight
		Value to Customer	40	Value to Business	20	Minimise Implementation Cost	10	Ease of Implementation	30		
Requirement 1	1	2	0.8	7	1.4	3	0.3	8	2.4	4.9	
Requirement 2	2	8	3.2	8	1.6	5	0.5	7	2.1	7.4	
Requirement 3	3	7	2.8	3	0.6	7	0.7	4	1.2	5.3	
Requirement 4	4	6	2.4	8	1.6	3	0.3	5	1.5	5.8	
Requirement 5	5	5	2	5	1	1	0.1	3	0.9	4	
Requirement 6	6	9	4	6	1.2	6	0.6	5	1.5	6.9	
Requirement 7	7	4	2	3	0.6	6	0.6	7	2.1	4.9	

"Value" Value to business Ease of implementation
Quantifiable indicators



Risk and Risk Management

- What are some categories of risk to a software project?

Too small at budget (cost)
Solving wrong problem
(customer related)
Not enough time (schedule)
Technology

Categories of Risk

- Product size
- Business impact
- Customer-related
- Process —
- Technology —
- Development environment —
- Staffing (size and experience) —
- Schedule —
- Cost

Top 10 Software Risks

- 10. Straining computer science capabilities -
- 9. Real-time performance shortfalls -
8. Shortfalls in externally performed tasks -
7. Shortfalls in externally furnished components -
6. Continuing stream of requirement changes -
5. Gold plating -
4. Developing wrong user interface -
3. Developing the wrong software functions
2. Unrealistic schedules and budgets -
1. Personnel shortfalls -

Hub Search

Boehm's Top 10 2002 - Software Risks

1. Schedules, budgets, process
2. Requirements Changes
3. Personnel Shortfalls
4. Requirements Mismatch
5. Rapid change
6. Architecture, performance, quality, distribution / mobility
7. COTS, external components
8. Legacy Software
9. Externally-performed tasks
10. User interface mismatch

*Contractors /
outsourced*

Problem.

Managing Risk

- Identify the risk —
- Estimate the damage from the risk (severity) —
- Estimate the probability of failure *How likely to occur*
- Determine the risk exposure
 - Product of the two

