

University of Toronto Department of Computer Science

Lecture 4, Part 1: the Feasibility Study

- ⇒ What is a feasibility study?
 - ↳ What to study and conclude?
- ⇒ Types of feasibility
 - ↳ Technical
 - ↳ Economic
 - ↳ Schedule
 - ↳ Operational
- ⇒ Quantifying benefits and costs
 - ↳ Payback analysis
 - ↳ Net Present Value Analysis
 - ↳ Return on Investment Analysis
- ⇒ Comparing alternatives

© Easterbrook 2004 1

University of Toronto Department of Computer Science

Content of a feasibility study

- ⇒ Things to be studied in the feasibility study:
 - ↳ The present organizational system
 - ↳ Stakeholders, users, policies, functions, objectives,...
 - ↳ Problems with the present system
 - ↳ inconsistencies, inadequacies in functionality, performance,...
 - ↳ Goals and other requirements for the new system
 - ↳ Which problem(s) need to be solved?
 - ↳ What would the stakeholders like to achieve?
 - ↳ Constraints
 - ↳ including nonfunctional requirements on the system (preliminary pass)
 - ↳ Possible alternatives
 - ↳ "Sticking with the current system" is always an alternative
 - ↳ Different business processes for solving the problems
 - ↳ Different levels/types of computerization for the solutions
 - ↳ Advantages and disadvantages of the alternatives
- ⇒ Things to conclude:
 - ↳ Feasibility of the project
 - ↳ The preferred alternative.

© Easterbrook 2004 3

University of Toronto Department of Computer Science

Why a feasibility study?

- ⇒ Objectives of a feasibility study:
 - ↳ To find out if an system development project can be done:
 - ↳ ...is it possible?
 - ↳ ...is it justified?
 - ↳ To suggest possible alternative solutions.
 - ↳ To provide management with enough information to know:
 - ↳ Whether the project can be done
 - ↳ Whether the final product will benefit its intended users
 - ↳ What the alternatives are (so that a selection can be made in subsequent phases)
 - ↳ Whether there is a preferred alternative
- ⇒ A feasibility study is a management-oriented activity
 - ↳ After a feasibility study, management makes a "go/no-go" decision.
 - ↳ Need to examine the problem in the context of broader business strategy

© Easterbrook 2004 2

University of Toronto Department of Computer Science

Exploring Feasibility

- ⇒ The "PIECES" framework
 - ↳ Useful for identifying operational problems to be solved, and their urgency
 - ↳ **P**erformance
 - ↳ Is current throughput and response time adequate?
 - ↳ **I**nformation
 - ↳ Do end users and managers get timely, pertinent, accurate and usefully formatted information?
 - ↳ **E**conomy
 - ↳ Are services provided by the current system cost-effective?
 - ↳ Could there be a reduction in costs and/or an increase in benefits?
 - ↳ **C**ontrol
 - ↳ Are there effective controls to protect against fraud and to guarantee information accuracy and security?
 - ↳ **E**fficiency
 - ↳ Does current system make good use of resources: people, time, flow of forms,...
 - ↳ **S**ervices
 - ↳ Are current services reliable? Are they flexible and expandable?

See the course website for a more specific list of PIECES questions

© Easterbrook 2004 4

University of Toronto Department of Computer Science

Four Types of feasibility

- ⇒ **Technical feasibility**
 - ↳ Is the project possible with current technology?
 - How much technical risk is there?
 - ↳ Does the technology exist at all?
 - Is it available locally?
 - Can it be obtained?
 - Will it be compatible with other systems?
- ⇒ **Economic feasibility**
 - ↳ Is the project possible, given resource constraints?
 - ↳ What benefits will result from the system?
 - Both tangible and intangible benefits
 - Quantify them!
 - ↳ What are the development and operational costs?
 - ↳ Are the benefits worth the costs?
- ⇒ **Schedule feasibility**
 - ↳ Is it possible to build a solution in time to be useful:
 - Any constraints on the schedule?
 - Can these constraints be met?
- ⇒ **Operational feasibility**
 - ↳ Urgency of the problem and the acceptability of any solution:
 - If the system is developed, will it be used?
 - ↳ Human and social issues...
 - ↳ internal issues:
 - Available of human resources?
 - Potential labour objections?
 - Manager resistance?
 - Organizational conflicts and policies?
 - ↳ external issues:
 - Social acceptability?
 - legal aspects and government regulations?

© Easterbrook 2004 5

University of Toronto Department of Computer Science

Economic Feasibility

- ⇒ **Can the bottom line be quantified yet?**
 - ↳ Very early in the project...
 - a judgement of whether solving the problem is worthwhile.
 - ↳ Once specific requirements and solutions have been identified...
 - ...the costs and benefits of each alternative can be calculated
- ⇒ **Cost-benefit analysis**
 - ↳ Purpose - answer questions such as:
 - Is the project justified (i.e. will benefits outweigh costs)?
 - Can the project be done, within given cost constraints?
 - What is the minimal cost to attain a certain system?
 - Which alternative offers the best return on investment?
 - ↳ Examples of things to consider:
 - Hardware/software selection
 - How to convince management to develop the new system
 - Selection among alternative financing arrangements (rent/lease/purchase)
 - ↳ Difficulties
 - benefits and costs can both be intangible, hidden and/or hard to estimate
 - ranking multi-criteria alternatives

© Easterbrook 2004 7

University of Toronto Department of Computer Science

Technical Feasibility

- ⇒ **Is the proposed technology or solution practical?**
 - ↳ Do we currently possess the necessary technology?
 - ↳ Do we possess the necessary technical expertise, and is the schedule reasonable?
 - ↳ Is relevant technology mature enough to be easily applied to our problem?
- ⇒ **What kinds of technology will we need?**
 - ↳ Some organizations like to use state-of-the-art technology
 - ...but most prefer to use mature and proven technology.
 - ↳ A mature technology has a larger customer base for obtaining advice concerning problems and improvements.
- ⇒ **Is the required technology available "in house"?**
 - ↳ If the technology is available:
 - ...does it have the capacity to handle the solution?
 - ↳ If the technology is not available:
 - ...can it be acquired?

© Easterbrook 2004 6

University of Toronto Department of Computer Science

Benefits and Costs

- ⇒ **Tangible Benefits**
 - ↳ Readily quantified as \$ values
 - ↳ Examples:
 - increased sales
 - cost/error reductions
 - increased throughput/efficiency
 - increased margin on sales
 - more effective use of staff time
- ⇒ **Intangible benefits**
 - ↳ Difficult to quantify
 - But maybe more important!
 - business analysts help estimate \$ values
 - ↳ Examples:
 - increased flexibility of operation
 - higher quality products/services
 - better customer relations
 - improved staff morale
- ⇒ **How will the benefits accrue?**
 - ↳ When - over what timescale?
 - ↳ Where in the organization?
- ⇒ **Development costs (OTO)**
 - ↳ Development and purchasing costs:
 - Cost of development team
 - Consultant fees
 - software used (buy or build)?
 - hardware (what to buy, buy/lease)?
 - facilities (site, communications, power,...)
 - ↳ Installation and conversion costs:
 - installing the system,
 - training personnel,
 - file conversion,....
- ⇒ **Operational costs (on-going)**
 - ↳ System Maintenance:
 - hardware (repairs, lease, supplies,...),
 - software (licenses and contracts),
 - facilities
 - ↳ Personnel:
 - For operation (data entry, backups,...)
 - For support (user support, hardware and software maintenance, supplies,...)
 - ↳ On-going training costs

© Easterbrook 2004 8

University of Toronto Department of Computer Science

Example: costs for small Client-Server project

Personnel:

2	System Analysts (400 hours/ea \$35.00/hr)	\$28,000
4	Programmer/Analysts (250 hours/ea \$25.00/hr)	\$25,000
1	GUI Designer (200 hours/ea \$35.00/hr)	\$7,000
1	Telecommunications Specialist (50 hours/ea \$45.00/hr)	\$2,250
1	System Architect (100 hours/ea \$45.00/hr)	\$4,500
1	Database Specialist (15 hours/ea \$40.00/hr)	\$600
1	System Librarian (250 hours/ea \$10.00/hr)	\$2,500

Expenses:

4	Smalltalk training registration (\$3500.00/student)	\$14,000
---	---	----------

New Hardware & Software:

1	Development Server (Pentium Pro class)	\$18,700
1	Server Software (operating system, misc.)	\$1,500
1	DBMS server software	\$7,500
7	DBMS Client software (\$950.00 per client)	\$6,650

Total Development Costs: **\$118,200**

PROJECTED ANNUAL OPERATING COSTS

Personnel:

2	Programmer/Analysts (125 hours/ea \$25.00/hr)	\$6,250
1	System Librarian (20 hours/ea \$10.00/hr)	\$200

Expenses:

1	Maintenance Agreement for Pentium Pro Server	\$995
1	Maintenance Agreement for Server DBMS software	\$525
1	Preprinted forms (15,000/year @ .22/form)	\$3,300

Total Projected Annual Costs: **\$11,270**

© Easterbrook 2004 9

University of Toronto Department of Computer Science

Calculating Present Value

- ⊖ A dollar today is worth more than a dollar tomorrow...
 - ⚡ Your analysis should be normalized to "current year" dollar values.
- ⊖ The discount rate
 - ⚡ measures opportunity cost:
 - > Money invested in this project means money not available for other things
 - > Benefits expected in future years are more prone to risk
 - ⚡ This number is company- and industry-specific.
 - > "what is the average annual return for investments in this industry?"
- ⊖ Present Value:
 - ⚡ The "current year" dollar value for costs/benefits n years into the future
 - > ... for a given discount rate i

$$\text{Present_Value}(n) = \frac{1}{(1+i)^n}$$

- ⚡ E.g. if the discount rate is 12%, then
 - > Present_Value(1) = $1/(1 + 0.12)^1 = 0.893$
 - > Present_Value(2) = $1/(1 + 0.12)^2 = 0.797$

© Easterbrook 2004 11

University of Toronto Department of Computer Science

Analyzing Costs vs. Benefits

- ⊖ Identify costs and benefits
 - ⚡ Tangible and intangible, one-time and recurring
 - ⚡ Assign values to costs and benefits
- ⊖ Determine Cash Flow
 - ⚡ Project costs and benefits over time, e.g. 3-5 years
 - ⚡ Calculate **Net Present Value** for all future costs/benefits
 - > determines future costs/benefits of the project in terms of today's dollar values
 - > A dollar earned today is worth more than a potential dollar earned next year
- ⊖ Do cost/benefit analysis
 - ⚡ Calculate **Return on Investment**:
 - > Allows comparison of lifetime profitability of alternative solutions.
 - $$\text{ROI} = \frac{\text{Lifetime benefits} - \text{Lifetime costs}}{\text{Lifetime costs}}$$
 - ⚡ Calculate **Break-Even point**:
 - > how long will it take (in years) to pay back the accrued costs:
 - $$\text{Accrued Cost (initial + incremental)} < \text{Accrued Benefit}$$

© Easterbrook 2004 10

University of Toronto Department of Computer Science

Net Present Value

- ⊖ Measures the total value of the investment
 - ⚡ ...with all figures adjusted to present dollar values
 - $$\text{NPV} = \text{Cumulative PV of all benefits} - \text{Cumulative PV of all costs}$$

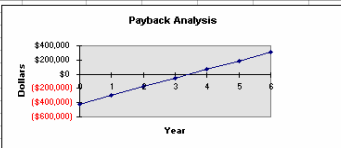
Cash Flow	Year 0	Year 1	Year 2	Year 3	Year 4
Dev. Costs	(\$100,000)				
Oper. Costs		(\$4,000)	(\$4,500)	(\$5,000)	(\$5,500)
Present Value	1	0.893	0.797	0.712	0.636
Time-adj Costs	(\$100,000)	(\$3,572)	(\$3,587)	(\$3,560)	(\$3,816)
Cumulative Costs	(\$100,000)	(\$103,572)	(\$107,159)	(\$110,719)	(\$114,135)
Benefits	0	\$25,000	\$30,000	\$35,000	\$50,000
T-adj Benefits	0	\$22,325	\$23,910	\$24,920	\$31,800
Cumulative Benefits	0	\$22,325	\$46,235	\$71,155	\$102,955
Net Costs+Benefits	(\$100,000)	(\$81,243)	(\$60,924)	(\$39,564)	(\$11,580)

- ⚡ Assuming subsequent years are like year 4...
 - > the net present value of this investment in the project will be:
 - > after 5 years, \$13,652
 - > after 6 years, \$36,168

© Easterbrook 2004 12



	A	B	C	D	E	F	G	H	I
1	Payback Analysis for Client-Server System Alternative								
2	(Numbers rounded to nearest \$1)								
3									
4	Cash flow description	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6	
5	Development cost:	(\$418,040)							
6	Operation & maintenance cost:		(\$15,045)	(\$16,000)	(\$17,000)	(\$18,000)	(\$19,000)	(\$20,000)	
7	Discount factors for 12%:	1.000	0.893	0.797	0.712	0.636	0.567	0.507	
8	Time-adjusted costs (adjusted to present)	(\$418,040)	(\$13,435)	(\$12,752)	(\$12,104)	(\$11,448)	(\$10,773)	(\$10,140)	
9	Cumulative time-adjusted costs over	(\$418,040)	(\$431,475)	(\$444,227)	(\$456,331)	(\$467,779)	(\$478,552)	(\$488,692)	
10									
11	Benefits derived from operation of new		\$0	\$150,000	\$170,000	\$190,000	\$210,000	\$230,000	\$250,000
12	Discount factors for 12%:	1.000	\$0.89	\$0.80	\$0.71	\$0.64	\$0.57	\$0.51	
13	Time-adjusted benefits (current of present)		\$0	\$133,950	\$135,490	\$135,260	\$133,560	\$130,410	\$126,750
14	Cumulative time-adjusted benefits over	\$0	\$133,950	\$269,440	\$404,720	\$538,280	\$668,690	\$795,440	
15	Cumulative lifetime time-adjusted costs +	(\$418,040)	(\$297,525)	(\$174,787)	(\$51,611)	\$70,501	\$190,138	\$306,748	



Return on Investment (ROI) analysis

For comparing overall profitability

- Which alternative is the best investment?
- ROI measures the ratio of the value of an investment to its cost.

ROI is calculated as follows:

$$\text{ROI} = \frac{\text{Estimated lifetime benefits} - \text{Estimated lifetime costs}}{\text{Estimated lifetime costs}}$$

or:

$$\text{ROI} = \frac{\text{Net Present value}}{\text{Estimated lifetime costs}}$$

For our example

- ROI = $(795,440 - 488,692) / 488,692 = 62.76\%$,
- or ROI = $306,748 / 488,692 = 62.76\%$

Solution with the highest ROI is the best alternative

- But need to know payback period too to get the full picture
- E.g. A lower ROI with earlier payback may be preferable in some circumstances



Computing the payback period

Can compute the break-even point:

- when does lifetime benefits overtake lifetime costs?
- Determine the fraction of a year when payback actually occurs:

$$\frac{\text{beginningYear amount}}{\text{endYear amount} + \text{beginningYear amount}}$$

- For our last example, $51,611 / (70,501 + 51,611) = 0.42$
- Therefore, the payback period is 3.42 years



Schedule Feasibility

How long will it take to get the technical expertise?

- We may have the technology, but that doesn't mean we have the skills required to properly apply that technology.

- May need to hire new people
- Or re-train existing systems staff
- Whether hiring or training, it will impact the schedule.

Assess the schedule risk:

- Given our technical expertise, are the project deadlines reasonable?
- If there are specific deadlines, are they mandatory or desirable?
 - If the deadlines are not mandatory, the analyst can propose several alternative schedules.

What are the real constraints on project deadlines?

- If the project overruns, what are the consequences?
 - Deliver a properly functioning information system two months late...
 - ...or deliver an error-prone, useless information system on time?
- Missed schedules are bad, but inadequate systems are worse!

University of Toronto Department of Computer Science

Operational Feasibility

- ⇒ How do end-users and managers feel about...
 - ↳ ...the problem you identified?
 - ↳ ...the alternative solutions you are exploring?
- ⇒ You must evaluate:
 - ↳ Not just whether a system *can* work...
 - ↳ ... but also whether a system *will* work.
- ⇒ Any solution might meet with resistance:
 - ↳ Does **management** support the project?
 - ↳ How do the **end users** feel about their role in the new system?
 - ↳ Which users or managers may resist (or not use) the system?
 - > People tend to resist change.
 - > Can this problem be overcome? If so, how?
 - ↳ How will the working environment of the end users change?
 - ↳ Can or will end users and management adapt to the change?

© Easterbrook 2004 17

University of Toronto Department of Computer Science

Comparing Alternatives

- ⇒ How do we compare alternatives?
 - ↳ When there are multiple selection criteria?
 - ↳ When none of the alternatives is superior across the board?
- ⇒ Use a Feasibility Analysis Matrix!
 - ↳ The columns correspond to the candidate solutions;
 - ↳ The rows correspond to the feasibility criteria;
 - ↳ The cells contain the feasibility assessment notes for each candidate;
 - ↳ Each row can be assigned a rank or score for each criterion
 - > e.g., for operational feasibility, candidates can be ranked 1, 2, 3, etc.
 - ↳ A final ranking or score is recorded in the last row.
- ⇒ Other evaluation criteria to include in the matrix
 - ↳ quality of output
 - ↳ ease of use
 - ↳ vendor support
 - ↳ cost of maintenance
 - ↳ load on system

© Easterbrook 2004 19

University of Toronto Department of Computer Science

Feasibility Study Contents

1. Purpose & scope of the study
 - ↳ Objectives (of the study)
 - ↳ who commissioned it & who did it,
 - ↳ sources of information,
 - ↳ process used for the study,
 - ↳ how long did it take,...
2. Description of present situation
 - ↳ organizational setting, current system(s).
 - ↳ Related factors and constraints.
3. Problems and requirements
 - ↳ What's wrong with the present situation?
 - ↳ What changes are needed?
4. Objectives of the new system.
 - ↳ Goals and relationships between them
5. Possible alternatives
 - ↳ ...including 'do nothing'.
6. Criteria for comparison
 - ↳ definition of the criteria
7. Analysis of alternatives
 - ↳ description of each alternative
 - ↳ evaluation with respect to criteria
 - ↳ cost/benefit analysis and special implications.
8. Recommendations
 - ↳ what is recommended and implications
 - ↳ what to do next;
 - > E.g. may recommend an interim solution and a permanent solution
9. Appendices
 - ↳ to include any supporting material.

© Easterbrook 2004 18

University of Toronto Department of Computer Science

Example matrix

	Candidate 1 Name	Candidate 2 Name	Candidate 3 Name
Description			
Operational Feasibility			
Technical Feasibility			
Schedule Feasibility			
Economic Feasibility			
Ranking			

© Easterbrook 2004 20

University of Toronto Department of Computer Science

Feasibility Criteria	Wt.	Candidate 1	Candidate 2	Candidate 3	Candidate E
Operational Feasibility Functionality - Describes to what degree the alternative would benefit the organization and how well the system would work. Political - A description of how well received this solution would be from both user management, user, and organization perspective.	30%	Only supports Member Services requirements and current business processes would have to be modified to take advantage of software functionality. Score: 60	Fully supports user required functionality. Score: 100	Same as candidate 2. Score: 100	
Technical Feasibility Technology - An assessment of the maturity, availability (or ability to acquire), and desirability of the computer technology needed to support this candidate. Expertise - An assessment to the technical expertise needed to develop, operate, and maintain the candidate system.	30%	Current production release of Platinum Plus package is version 1.0 and has only been on the market for 6 weeks. Maturity of product is a risk and company charges an additional monthly fee for technical support. Required to hire or train C++ expertise to perform modifications for integration requirements. Score: 50	Although current technical staff has only Powerbuilder experience, the senior analysts who saw the MS Visual Basic demonstration and presentation, has agreed the transition will be simple and finding experienced VB programmers will be easier than finding Powerbuilder programmers and at a much cheaper cost. MS Visual Basic 5.0 is a mature technology based on version number. Score: 95	Although current technical staff is comfortable with Powerbuilder, management is concerned with recent acquisition of Powerbuilder by Sybase Inc. MS SQL Server is a current company standard and competes with SYBASE in the Client/Server DBMS market. Because of this we have no guarantee future versions of Powerbuilder will Nplay well0 with our current version SQL Server. Score: 60	
Ranking	100%	60.5	92	83.5	

© Easterbrook 2004 21

University of Toronto Department of Computer Science

Lecture 4, Part 2: Stakeholder Goals

- ⇒ **Boundaries**
 - ↳ Scoping the problem
- ⇒ **Stakeholders**
 - ↳ Identifying the problem owners
- ⇒ **Goals**
 - ↳ Identifying the success criteria
- ⇒ **Scenarios**
 - ↳ Using concrete examples to understand the problem

© Easterbrook 2004 23

University of Toronto Department of Computer Science

Feasibility Criteria	Wt.	Candidate 1	Candidate 2	Candidate 3	Candidate E
Operational Feasibility	30%	Score: 60	Score: 100	Score: 100	
Technical Feasibility	30%	Score: 50	Score: 95	Score: 100	
Economic Feasibility Cost to develop: Payback period (discounted): Net present value: Detailed calculations:	30%	Approximately \$350,000. Approximately 4.5 years. Approximately \$210,000. See Attachment A. Score: 60	Approximately \$418,040. Approximately 3.5 years. Approximately \$306,748. See Attachment A. Score: 85	Approximately \$400,000. Approximately 3.3 years. Approximately \$325,500. See Attachment A. Score: 90	
Schedule Feasibility An assessment of how long the solution will take to design and implement.	10%	Less than 3 months. Score: 95	9-12 months Score: 80	9 months Score: 85	
Ranking	100%	60.5	92	83.5	

© Easterbrook 2004 22

University of Toronto Department of Computer Science

Where do we start?

- ⇒ **Identify the problem**
 - ↳ what is the objective of the project?
 - ↳ the "vision" of those who are pushing for it?
 - e.g., "Meeting scheduling is too costly right now"
- ⇒ **Scope the problem**
 - ↳ given the vision, how much do we tackle?
 - e.g. "Build a system that schedules meetings", ...or...
 - e.g. "Build a system that maintains people's calendars" ...or...
- ⇒ **Choose a business process ?**
 - ↳ given the problem, what is the appropriate business process for solving it?
 - e.g. "Anyone who wants to schedule a meeting goes to the secretary, gives details and the secretary handles the rest", ...or...
 - e.g. "Anyone can submit a meeting request, participants are informed and a negotiation settles meeting details" ...or...
- ⇒ **Choose among alternatives ?**
 - ↳ Given a business process, what parts should be automated, and how?
 - e.g. "Computer takes in scheduling request details, outputs a solution" ...or...
 - e.g. "Solution arrived at interactively by secretary and computer" ...or...

© Easterbrook 2004 24

University of Toronto Department of Computer Science

Requirements Elicitation

- Starting point
 - Some notion that there is a "problem" that needs solving
 - e.g. dissatisfaction with the current state of affairs
 - e.g. a new business opportunity
 - e.g. a potential saving of cost, time, resource usage, etc.
- Collect enough information to:
 - identify the "problem"/"opportunity"
 - Which problem needs to be solved? (Identify problem **Boundaries**)
 - Where is the problem? (understand the **Context/Problem Domain**)
 - Whose problem is it? (Identify **Stakeholders**)
 - Why does it need solving? (Identify the stakeholders' **Goals**)
 - How does the problem manifest itself? (collect some **Scenarios**)
 - When does it need solving? (Identify **Development Constraints**)
 - What might prevent us solving it? (Identify **Feasibility and Risk**)
 - become an expert in the problem domain
 - Learn how to find your way round a new problem area quickly
 - Use your (initial) ignorance as an excuse to ask questions
 - Recognise the domain expertise of the people you talk to

W6H
The journalist's technique:
What?
Where?
Who?
Why?
When?
How?
(Which?)

© Easterbrook 2004 25

University of Toronto Department of Computer Science

Stakeholders

- Stakeholder analysis:
 - Identify all the people who must be consulted during information acquisition
- Example stakeholders
 - Users
 - concerned with the features and functionality of the new system
 - Designers
 - want to build a perfect system, or reuse existing code
 - Systems analysts
 - want to "get the requirements right"
 - Training and user support staff
 - want to make sure the new system is usable and manageable
 - Business analysts
 - want to make sure "we are doing better than the competition"
 - Technical authors
 - will prepare user manuals and other documentation for the new system
 - The project manager
 - wants to complete the project on time, within budget, with all objectives met.
 - "The customer"
 - Wants to get best value for money invested!

© Easterbrook 2004 27

University of Toronto Department of Computer Science

Identifying the Problem

- Vague problem stated by the customer:
 - E.g. university textbook store:
 - Manager wants to computerize the book order forms filled out by instructors;
 - E.g. A large insurance company:
 - Claims manager wants to cut down the average time it takes to process an insurance claim from 2 months to 2 weeks
 - E.g. A telecommunications company:
 - CIO wants to integrate the billing system with customer record systems of several affiliates, so there is only one billing system...
 - E.g. Large Government Aerospace Agency:
 - The president wants to send a manned mission to Mars by the the year 2020
- Often you only see symptoms rather than causes:
 - E.g. "Ontario patients needing X-ray scans have to wait for months"
 - The long wait is the symptom, not the problem. The problem may be:
 - Shortage of X-ray machines;
 - Shortage of trained staff;
 - Shortage of doctors to process the data
 - Inefficient scheduling procedures

© Easterbrook 2004 26

University of Toronto Department of Computer Science

Finding stakeholders: The Org Chart

```

graph TD
    Penny[Penny President] --> Oscar[Oscar Vice-President (Ops)]
    Penny --> Richard[Richard Vice-President (R&D)]
    Oscar --> Sally[Sally Sales Manager]
    Oscar --> Percy[Percy Production Manager]
    Richard --> Sarah[Sarah Chief Scientist]
    Richard --> Desmond[Desmond Head of Product Design]
    Sally --> S1[ ]
    Sally --> S2[ ]
    Percy --> P1[ ]
    Sarah --> S3[ ]
    Sarah --> S4[ ]
    Desmond --> D1[ ]
    Desmond --> D2[ ]
    D1 --> D1_1[ ]
    D1 --> D1_2[ ]
    D2 --> D2_1[ ]
    D2 --> D2_2[ ]
  
```

responsibility ↑

↓ authority

- Organization charts show
 - Areas of responsibility (flows upwards)
 - Lines of authority (delegated downwards)
- A useful tool for figuring out where the stakeholders are

© Easterbrook 2004 28

University of Toronto Department of Computer Science

Finding Stakeholders: Levels of authority

- Top management
 - establishes goals
 - does long-range planning
 - determines new market & product developments
 - decides on mergers & acquisitions.
- Middle management
 - sets objectives
 - allocates & controls resources
 - does planning
 - measures performance
- Lower management
 - supervises day-to-day operations
 - takes corrective action when necessary.
- Operational level
 - performs day-to-day operations

© Easterbrook 2004 29

University of Toronto Department of Computer Science

Goal Modeling

- (Hard) Goals:
 - Describe functions that must be carried out. E.g.
 - Satisfaction goals
 - Information goals
- Softgoals:
 - Cannot really be fully satisfied. E.g.
 - Accuracy
 - Performance
 - Security
 - ...
- Also classified temporally:
 - Achieve/Cease goals
 - Reach some desired state eventually
 - Maintain/Avoid goals
 - Keep some property invariant
 - Optimize
 - A criterion for selecting behaviours
- Agents:
 - Owners of goals
 - Choice of when to ascribe goals to agents:
 - Identify agents first, and then their goals
 - Identify goals first, and then allocate them to agents during operationalization
- Modelling Tips:
 - Multiple sources yield better goals
 - Associate stakeholders with each goal
 - reveals viewpoints and conflict
 - Use scenarios to explore how goals can be met
 - Explicit consideration of obstacles helps to elicit exceptions

© Easterbrook 2004 31

University of Toronto Department of Computer Science

Identifying Stakeholders' Goals

- Approach
 - Focus on why systems are constructed
 - Express the 'why' as a set of stakeholder goals
 - Use goal refinement to arrive at specific requirements
 - Goal analysis
 - document, organize and classify goals
 - Goal evolution
 - refine, elaborate, and operationalize goals
 - Goal hierarchies show refinements and alternatives
- Advantages
 - Reasonably intuitive
 - Explicit declaration of goals provides sound basis for conflict resolution
- Disadvantages
 - Captures a static picture - what if goals change over time?
 - Can regress forever up (or down) the goal hierarchy

© Easterbrook 2004 Source: Adapted from Anton, 1996. 30

University of Toronto Department of Computer Science

Example Goal Elaboration

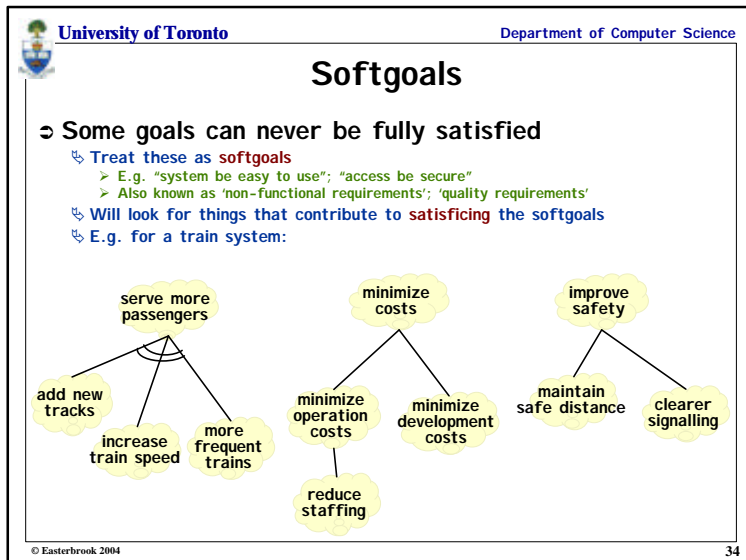
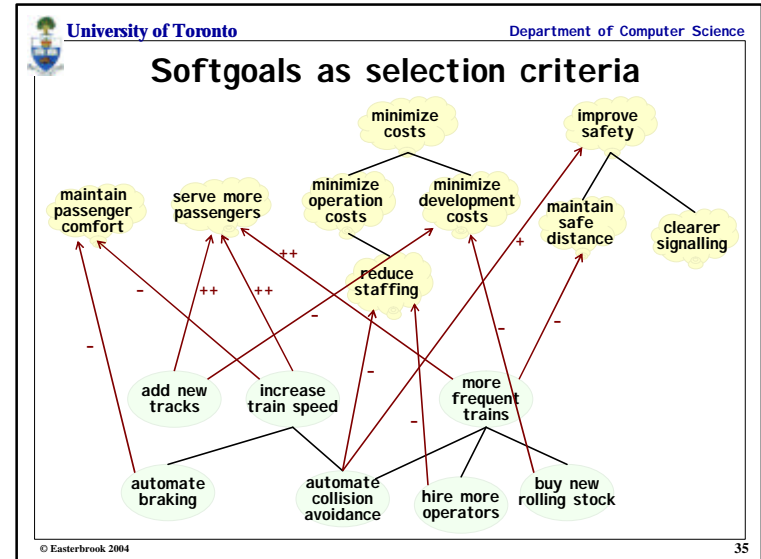
© Easterbrook 2004 32

University of Toronto Department of Computer Science

Goal Analysis

- Goal Elaboration:
 - "Why" questions explore higher goals (context)
 - "How" questions explore lower goals (operations)
 - "How else" questions explore alternatives
- Relationships between goals:
 - One goal **helps** achieve another (+)
 - One goal **hurts** achievement of another (-)
 - One goal **makes** another (++)
 - Achievement of one goal guarantees achievement of another
 - One goal **breaks** another (--)
 - Achievement of one goal prevents achievement of another
 - Precedence ordering - must achieve goals in a particular order
- Obstacle Analysis:
 - Can this goal be obstructed, if so how?
 - What are the consequences of obstructing it?

© Easterbrook 2004 33



University of Toronto Department of Computer Science

Scenarios

- Scenarios
 - Specific sequence of interaction between actor and system
 - Tend to be short (e.g between 3 and 7 steps)
 - May be:
 - positive (i.e. required behavior)
 - negative (i.e an undesirable interaction)
 - May be indicative (describe current system) or optative (how it should be)
- Advantages
 - Very natural: stakeholders tend to use them spontaneously
 - E.g "suppose I'm admitted to hospital - what happens during my admission?"
 - Typical answer: "You, or the person accompanying you would talk to the person at the admissions desk. You have to show your OHIP card and explain who referred you to the hospital. Then you..." [and so on]
 - Short scenarios very good for quickly illustrating specific interactions
- Disadvantages
 - Lack of structure:

© Easterbrook 2004 Source: Adapted from Dardenne, 1993. 36



Example Scenario

Title: Successful meeting scheduled using messaging option

Participants: Alice (initiator, not attending); Bob, Carlo, Daphne (attendees)

Action	Goals satisfied	Obstacles / Problems
Alice requests meeting, specifying participants, timeframe	Meeting requested; Attendee list obtained	What if selected timeframe is infeasible?
AS sends participant requests to Bob, Carlo and Daphne	?	Did we miss a goal?
Bob reads message	Participants informed	Can't detect when messages are read; what happens if Bob reads the message but doesn't reply?
Carlo reads message		
Daphne reads message		
Bob replies with preferences	Attendees preferences known	What if the preferences are mutually exclusive? Should we allow some to be higher priority?
Carlo replies with preferences		
Daphne replies with preferences		
AS schedules meeting	Room availability determined; room booked	
AS notifies Alice, Bob, Carlo, Daphne of time and location	Meeting announced; Attendance Confirmed (?)	How do we know if they've all read the announcement? What if the schedule is no longer convenient for one of them?