CSE 405
Software Engineering Economics

3. Risk and Uncertainty

Dr. Salisu Garba
Salisu.garba@slu.edu.ng

Summary

- Introduction to Risk and Uncertainty
- Goals, Estimates, and Plans
- Estimation Techniques
- Addressing Uncertainty
- Prioritization
- Decisions under Risk
- Decisions under Uncertainty

What Is Software Risk?

- A **possibility** of suffering from **loss** in software development process is called a software risk.
- Loss can be anything, increase in production **cost**, development of poor **quality** software, not being able to complete the project on **time**.
- Two characteristics of risk
  - **Uncertainty** – the risk **may or may not happen**, that is, there are no 100% risks (those, instead, are called constraints)
  - **Loss** – the **risk becomes a reality** and unwanted consequences or losses occur
- A software risk can be of two types
  1. Internal risks that are **within the control** of the **project manager** and
  2. External risks that are **beyond the control** of the **project manager**.

Risk Categorization

- **Project risks:** They threaten the **project plan**
  - If they become real, it is likely that the project **schedule will slip and that costs will increase**
- **Technical risks:** They threaten the **quality** and timeliness of the software to be produced
  - If they become real, **implementation** may become **difficult or impossible**
- **Business risks:** They threaten the **viability** of the software to be built
  - If they become real, they jeopardize the project or the product
  - **Market risk** – building an excellent product or system that **no one really wants**
  - **Strategic risk** – building a product that **no longer fits into the overall business strategy** for the company
  - **Sales risk** – building a product that the sales force **doesn't understand how to sell**
  - **Management risk** – **losing the support of senior management** due to a change in focus or a change in people
  - **Budget risk** – **losing budgetary** or personnel commitment

Known and Predictable Risk Categories

1-Product size	risks associated with overall size of the software to be built
2-Business impact	risks associated with constraints imposed by management or the marketplace
3-Customer characteristics	risks associated with sophistication of the customer and the developer's ability to communicate with the customer in a timely manner
4-Process definition	risks associated with the degree to which the software process has been defined and is followed
5-Development environment	risks associated with availability and quality of the tools to be used to build the project
6-Technology to be built	risks associated with complexity of the system to be built and the "newness" of the technology in the system
7-Staff size and experience	risks associated with overall technical and project experience of the software engineers who will do the work

Assessing Risk Impact

- Three factors affect the consequences that are likely if a risk does occur
  - **Its nature** – This indicates the **problems** that are likely if the risk occurs
  - **Its scope** – This combines the **severity** of the risk (how serious was it) with its overall distribution (how much was affected)
  - **Its timing** – This considers **when** and for **how long** the impact will be felt
- The overall **risk exposure** formula is **RE = P x C**
  - P = the **probability** of occurrence for a risk
  - C = the **cost** to the project should the risk actually occur
- Example
  - P = 80% probability that 18 of 60 software components will have to be re-developed
  - C = Total cost of developing 18 components is $25,000
  - RE = 0.80 x $25,000 = **$20,000**

Seven Principles of Risk Management

Maintain a global perspective	View software risks within the context of a system and the business problem that is intended to solve
Take a forward-looking view	Think about risks that may arise in the future; establish contingency plans
Encourage open communication	Encourage all stakeholders and users to point out risks at any time
Integrate risk management	Integrate the consideration of risk into the software process
Emphasize a continuous process of risk management	Modify identified risks as more becomes known and add new risks as better insight is achieved
Develop a shared product vision	A shared vision by all stakeholders facilitates better risk identification and assessment
Encourage teamwork when managing risk	Pool the skills and experience of all stakeholders when conducting risk management activities

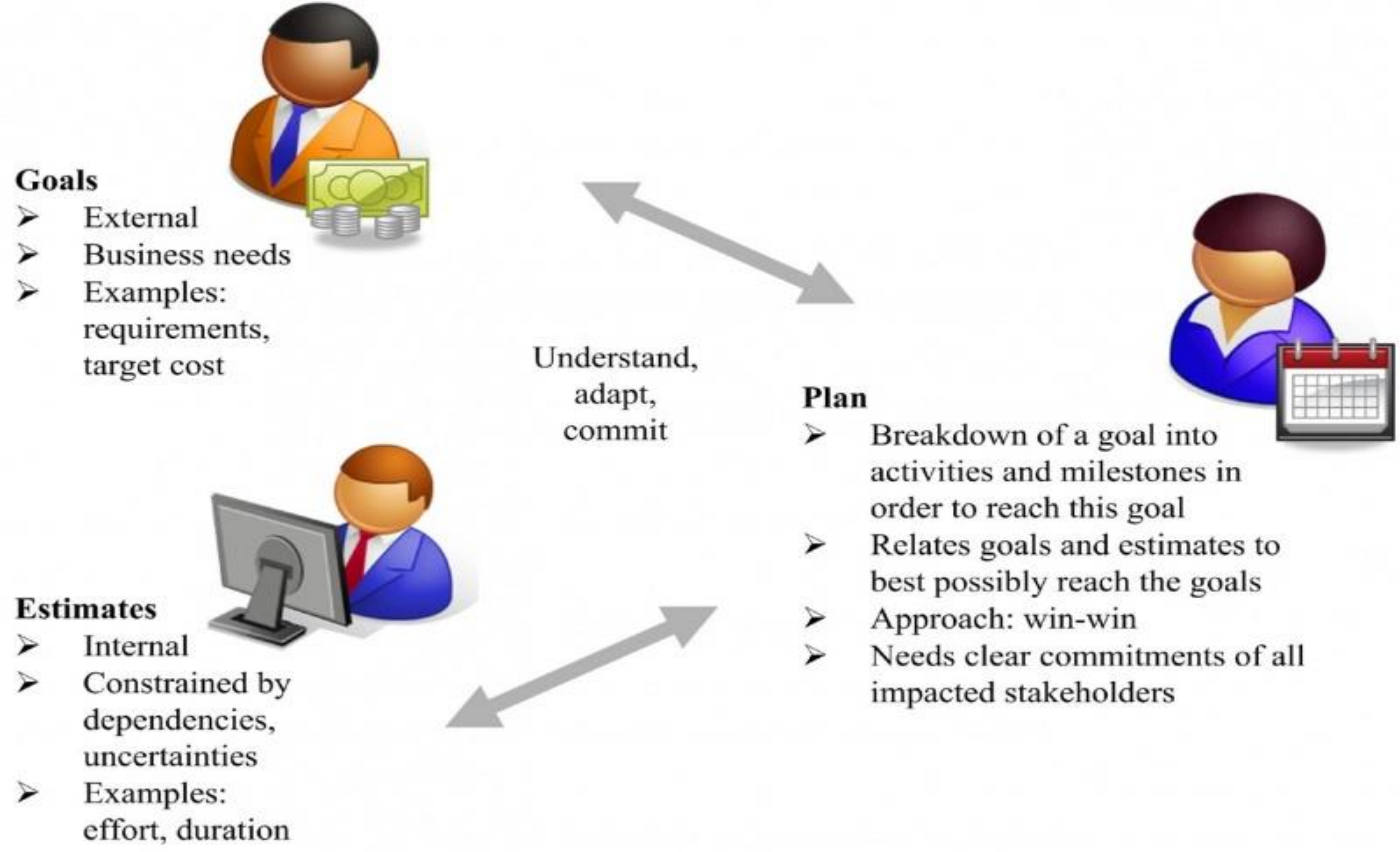
Goals, Estimates, and Plans

- **Goals** in software engineering economics are mostly **business goals** (or business objectives). A business goal relates business needs (such as **increasing profitability**) to investing resources (such as starting a project or launching a product with a given budget, content, and timing).

- **An estimate** is a well-founded **evaluation of resources and time** that will be needed to achieve stated goals. A software estimate is used to determine **whether the project goals can be achieved** within the constraints on **schedule, budget, features, and quality attributes**.

- **A plan** describes the **activities and milestones** that are necessary **in order to reach the goals of a project**. The plan should be in line with the goal and the estimate, which is not necessarily easy and obvious

Goals, Estimates, and Plans



Estimation Techniques

- Estimations are used to **analyze and forecast the resources or time** necessary to implement requirements. Five families of estimation techniques exist:
    1. **Expert judgment:** Individuals or organizations with particular knowledge or training in team and resource planning and estimating should be taken into consideration.
    2. **Analogy:** data collected from previous similar projects and compares them to your current project status to estimate the cost, resources, and duration.
    3. **Estimation by parts:** The system being costed is broken down into lower-level components, each of which is costed separately for direct labor, direct material, etc.
    4. **Parametric methods:** The parametric technique uses regression or other statistical methods to model complex projects
    5. **Statistical methods:** This method uses statistical data about the previous experience in the field in general
- No single estimation technique is perfect, so using multiple estimation technique is useful.

Introduction to COCOMO

- When Barry Boehm wrote 'Software Engineering Economics', published in 1981, he introduced an empirical effort **estimation model** that is still referenced by the software engineering community.

- The **COnstructive COst MOdel** (COCOMO) is the **most widely used** software estimation model in the world.

- The COCOMO model predicts the **effort** and **duration** of a project based on inputs relating to the **size** of the resulting systems and a number of "**cost drives**" that affect productivity.

COCOMO Models

- **Basic** - predicted software size (lines of code) was used to estimate development effort.
- **Intermediate** - predicted software size (lines of code), plus a set of 15 subjectively assessed 'cost drivers' was used to estimate development effort.
- **Advanced** - on top of the intermediate model, the advanced model allows phase-based cost driver adjustments and some adjustments at the module, component, and system level.
- The more complex models account for more factors that influence software projects, and make more accurate estimates.

The Development mode

- The most important factors contributing to a project's duration and cost is the Development Mode
  - **Organic Mode:** The project is developed in a familiar, stable environment, and the product is similar to previously developed products. The product is relatively small, and requires little innovation.
  - **Semidetached Mode:** The project's characteristics are intermediate between Organic and Embedded.
  - **Embedded Mode:** The project is characterized by tight, inflexible constraints and interface requirements. An embedded mode project will require a great deal of innovation.

Basic COCOMO model

The general formula of the basic COCOMO model is: $$E = a(S)^b$$

E represents effort in person-months,

S is the size of the software development in KLOC (1000LOC),

a and **b** are constant values dependent on the development mode,

Mode	a	b
Organic	2.4	1.05
Semi-detached	3.0	1.12
Embedded	3.6	1.20

Example

Mode	Effort Formula
Organic	$E = 2.4 * (S^{1.05})$
Semidetached	$E = 3.0 * (S^{1.12})$
Embedded	$E = 3.6 * (S^{1.20})$

$\text{Size} = 200 \text{ KLOC}$

$\text{Effort} = a * \text{Size}^{b}$

Organic — $E = 2.4 * (200^{1.05}) = 626$ staff-months

Semidetached — $E = 3.0 * (200^{1.12}) = 1133$ staff-months

Embedded — $E = 3.6 * (200^{1.20}) = 2077$ staff-months

The intermediate & advanced COCOMO

The intermediate and advanced COCOMO models incorporate 15 'cost drivers'. These 'drivers' multiply the effort derived for the basic COCOMO model. The importance of each driver is assessed and the corresponding value multiplied into the COCOMO equation, which becomes: $\mathbf{E = a(S)^b \text{ x } product(cost\ drivers)}$

Mode	a	b
Organic	3.2	1.05
Semi-detached	3.0	1.12
Embedded	2.8	1.20

COCOMO Cost Drivers

COST DRIVER	DESCRIPTION
RELY	Required software reliability
DATA	Database size
CPLX	Product complexity
TIME	Execution time constraints
STOR	Main storage constraints
VIRT	Virtual machine volatility - degree to which the operating system changes
TURN	Computer turn around time
ACAP	Analyst capability
AEXP	Application experience
PCAP	Programmer capability
VEXP	Virtual machine (i.e. operating system) experience
LEXP	Programming language experience
MODP	Use of modern programming practices
TOOL	Use of software tools
SCED	Required development schedule

The cost drivers table

Cost Driver	Very Low	Low	Nominal	High	Very High	Extra High
Required Reliability (PRODUCT)	.75	.88	1.00	1.15	1.40	1.40
Database Size (PRODUCT)	.94	.94	1.00	1.08	1.16	1.16
Product Complexity (PRODUCT)	.70	.85	1.00	1.15	1.30	1.65
Execution Time Constraint (COMP.)	1.00	1.00	1.00	1.11	1.30	1.66
Main Storage Constraint (COMP.)	1.00	1.00	1.00	1.06	1.21	1.56
Virtual Machine Volatility (COMP.)	.87	.87	1.00	1.15	1.30	1.30
Comp Turn Around Time (COMP.)	.87	.87	1.00	1.07	1.15	1.15
Analyst Capability (PERSONNEL)	1.46	1.19	1.00	.86	.71	.71
Application Experience (PERSON.)	1.29	1.13	1.00	.91	.82	.82
Programmers Capability (PERSON.)	1.42	1.17	1.00	.86	.70	.70
Virtual machine Experience (PERS.)	1.21	1.10	1.00	.90	.90	.90
Language Experience (PERSONNEL)	1.14	1.07	1.00	.95	.95	.95
Modern Prog Practices (PROJECT)	1.24	1.10	1.00	.91	.82	.82
SW Tools (PROJECT)	1.24	1.10	1.00	.91	.83	.83
Required Dev Schedule (PROJECT)	1.23	1.08	1.00	1.04	1.10	1,10

Example

- The following is a calculation of the estimated effort for a semi-detached project of 56 KLOC. The cost drivers are set as follows:

Required Reliability	High = 1.15 (from the table)
Database Size	Low = 0.94 (from the table)
Product Complexity	Normal = 1.00 (from the table)
SW Tools	Very low = 1.24 (from the table)

hence, product(cost drivers) = 1.15 x 0.94 x 1.00 x 1.24 = **1.34044**

for a semi-detached project of 56KLOC: a = 3.0 b = 1.12 S = 56

$$E = a(S)^b \times \text{product(cost drivers)}$$

$$E = 3.0 \times (56)^{1.12} \times 1.34044$$

E = 3.0 X 90.78 X 1.34044

E = 365.0554 Person-month

Software Development Time

- **Development Time Equation Parameter Table:**

Parameter	Organic	Semi-detached	Embedded
C	2.5	2.5	2.5
D	0.38	0.35	0.32

Development-time, **TDEV = C * (E)^D**

Development-time = $2.5(365.0554)^{0.35}$ =19.7136 Months

Number of Personnel, **NP = E / TDEV**

NP = 365.0554PM / 19.7136 Months = 19 People

Addressing Uncertainty

- Because of the many **unknown factors** during project initiation and planning, **estimates are inherently uncertain**; that uncertainty should be addressed in business decisions.
- **Techniques for addressing uncertainty includes**:
  - Consider ranges of estimates
  - Analyze **sensitivity** to changes of assumptions
  - Delay final decisions.

Sensitivity Analysis

- A technique used to study how **various sources of uncertainty** in a mathematical model contribute to the model's overall uncertainty.

- A systematic way of asking **"what-if" scenario questions** in order to understand:
  - What outcomes could possibly occur that would effect your decision
  - Which input assumptions have the most impact on key outputs and thus your decision and should be analyzed further
- Performed by setting all inputs equal to their base values and changing one variable's assumed value at a time
- The following are some advantages of sensitivity analysis:
  - #1. It Examines Many Scenarios
  - #2. Enhanced Managerial Judgment
  - #3. Effective Resource Management
  - #5. Provides a Higher Level of Credibility

Sensitivity Analysis (Step by Step)

- Firstly, the analyst is required to design the **basic formula**, which will act as the **output formula**. For instance, say NPV formula can be taken as the output formula.
- Next, the analyst needs to identify which are the **variables that are required to be sensitized** as they are key to the output formula. In the NPV formula in excel, the cost of capital and the initial investment can be the independent variables.
- Next, determine the **probable range** of the independent variables.
- Next, open an excel sheet and then put the range of one of the independent variable along the rows and the other set along with the columns.

Sensitivity Analysis Example 1

- Let us take the example of a simple output formula, which is stated as the summation of the **square of two independent variables X and Y**.
- In this case, let us assume the range of X as 2, 4, 6, 8, and 10, while that of Y as 1, 3, 5, 7, 9, 11, and 13. Based on the above-mentioned technique, all the combinations of the two independent variables will be calculated to assess the sensitivity of the output.
- For instance, if X = 2 (Cell B2) and Y = 7 (Cell B3), then Z = $2^2 + 7^2$ = 53 (Cell B4)

SUM | =B3^2+B4^2

	A	B	C	D
3	X	2		
4	Y	7		
5	Z	=B3^2+B4^2		
6				

	A	B
3	X	2
4	Y	7
5	Z	53

	A	B	C	D	E	F	G
7			X				
8		53	2	4	6	8	10
9		1	5	17	37	65	101
10		3	13	25	45	73	109
11		5	29	41	61	89	125
12	Y	7	53	65	85	113	149
13		9	85	97	117	145	181
14		11	125	137	157	185	221
15		13	173	185	205	233	269

Sensitivity Analysis Example 2

- Example of how to build a sensitivity analysis table in Excel to evaluate how **changing two inputs simultaneously** will affect an output (**profit**)

Assumptions		
Chairs Sold		1,000
Price / Chair		$ 150
Cost / Chair		$ 50
Store Rent		$ 10,000
Payroll		$ 50,000

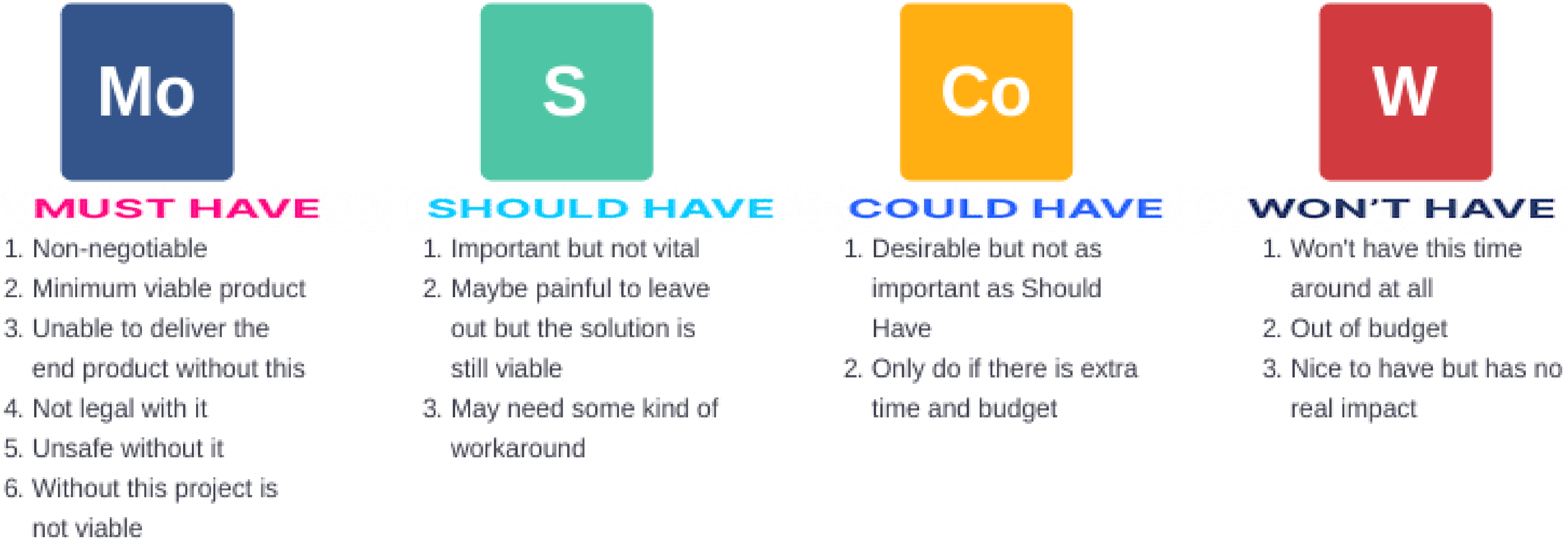
Profit & Loss Statement		
Revenue	Chair Sold X Price	$150,000
Cost of Sales	Chair Sold X Cost	$ 50,000
Gross Profit	Revenue - Cost of Sale	$100,000
SG&A	Rent + Payroll	$ 60,000
Operating Profit	Gross Profit - SG&A	$ 40,000

		No. of Chairs Sold				
	$40,000	500	750	1000	1250	1500
Price	$ 200	15000	52500	90000	127500	165000
	$ 175	2500	33750	65000	96250	127500
	$ 150	-10000	15000	40000	65000	90000
	$ 125	-22500	-3750	15000	33750	52500
	$ 100	-35000	-22500	-10000	2500	15000
	$ 75	-47500	-41250	-35000	-28750	-22500

SG&A =Selling, General & Administrative Expenses

Prioritization

- Prioritization involves **ranking alternatives based on common criteria** to deliver the **best possible value**. In software engineering projects, software requirements are often prioritized in order to deliver the most value to the client within constraints of schedule, budget, resources, and technology.



Prioritization Example

- Imagine that **you have been hired to create a Website** for a Law firm.
- They want **a professional Site** where **people can Register** and, **once inside, track their court cases**.
- Since you want to deliver the best possible Site on time, **you decide to follow the MoSCoW method**.

Must Have:
Solid programming without any bugs.
A Solid Register System.
A Safe and Reliable personal directory.

Should Have:
A Fast Site.
An outstanding Design.
Notifications sent by e-mail.

Could Have:
Custom menus.
Suggestions.
A Blog section with latest news.

Won't Have:
Paid content.
A Public Members section.
Ads.

Decisions Under Risk

- Decisions under risk techniques are used when the decision maker can **assign probabilities** to the **different possible outcomes**. The specific techniques include:
  - Expected value decision making
  - Expectation variance and decision making
  - Monte Carlo analysis
  - Expected value of perfect information.

Expected Value Decision Making

- Expected value allows decision-makers to **evaluate potential scenarios and outcomes**, & make decisions based on the probability of success.
- The formula for calculating expected value looks like this:

$$\text{Expected Value} = \sum_{i=1}^{n} (\text{Value}_i \times \text{Probability}_i)$$

- **EV** = (Probability of Outcome A x Value of Outcome A) + (Probability of Outcome B x Value of Outcome B) + (Probability of Outcome C x Value of Outcome C) + (Probability of Outcome D x Value of Outcome D)
- For example, if there are four possible outcomes with values of $20, $30, $40, and $50 respectively and the associated probabilities are 0.2, 0.3, 0.2, and 0.3 respectively, the expected value is calculated as follows:
- **EV** = (0.2 x $20) + (0.3 x $30) + (0.2 x $40) + (0.3 x $50) = $34

Expected Value Example

- Imagine 1000 parallel universities where the HRM System could be run at the same time
  - Most favorable outcome would happen in 150 of those universes
  - Fair outcome would happen in 650
  - Least favorable outcome would happen in 200
- Several projects at HRM Systems

	Least Favorable	Fair	Most Favorable
ALTERNATIVES	20%	65%	15%
FUD	-$1234	$5678	$9012
SLU	-2101	6601	9282
KUST	-3724	4104	9804

- The concept can help when choosing from among several alternatives
  - FUD (0.20 * -$1234) + (0.65 * $5678) + (0.15 * $9012) = $4795.70
  - SLU (0.20 * -$2101) + (0.65 * $6601) + (0.15 * $9282) = **$5262.75**
  - KUST (0.20 * -$3724) + (0.65 * $4104) + (0.15 * $9804) = $3393.40

Expectation Variance & Decision Making

- What if probabilities were different for each alternative?

SLU

Outcome	Probability	
Least favorable	45%	-$3494
Nominal	10%	728
Most favorable	45%	4811

FUD

Outcome	Probability	
Least favorable	10%	-$200
Low nominal	20%	108
High nominal	30%	378
Most favorable	40%	877

SLU EXPECTED VALUE = **$665**

FUD EXPECTED VALUE = **$466**

- SLU has higher expected value but **win big - lose big**
- FUD has lower expected value but **more probability of profit**

Monte Carlo Analysis

- The Monte Carlo Analysis is a risk management technique, which project managers use to estimate the **impacts of various risks** on the project **cost** and project **timeline**.
- Randomly generate combinations of input values and look at distribution of outcomes
  - Named after gambling resort in Monaco

	Least favorable estimate	Fair estimate	Most favorable estimate
Initial investment	$500,000	$400,000	$360,000
Operating & maintenance	$1500	$1000	$800
Development staff cost / month	$49,000	$35,000	$24,500
Development project duration	15 months	10 months	7 months
Income / month	$24,000	$40,000	$56,000

Expected Value of Perfect Information

- The **expected value of perfect information** (**EVPI**), refers to the impact on **decisions when more information is provided** so that the decision-maker can reach a point of certainty.
- EVPI provides a reference for decision-makers to understand how much to spend for additional information.
- This additional data, whether it applies to understanding the market better or understanding the costs of making changes in the company, can help the decision-maker by providing this insight.
- **EVPI = EPPI - EMV**
- EPPI represents the expected payoff with perfect information while EMV stands for expected monetary value.

Decisions Under Uncertainty

- Decisions under uncertainty techniques are used when the decision maker **cannot assign probabilities to the different possible outcomes** because needed information is not available. Specific techniques include:
  - Laplace Rule
  - Maximin Rule
  - Maximax Rule
  - Hurwicz Rule
  - Minimax Regret Rule

Payoff/Decision Table

State of Nature/Outcomes
Decision maker has no control over them

	Economy		
Alternatives	**Growing**	**Stable**	**Declining**
Bonds	40	45	5
Stocks	70	30	-13
Mutual Funds	53	45	-5

Decision alternatives/options available for the decision maker

Payoffs (e.g profit, cost, etc.)

Maximax and Maximin Rule

Maximax Approach

- Maximax approach is also Optimistic approach
- It Maximizes the maximum payoff (Best of the Best)

	Economy			
Alternatives	Growing	Stable	Declining	Best
Bonds	40	45	5	45
Stocks	70	30	-13	70
Mutual Funds	53	45	-5	53

Decision: Invest in ***Stocks***

Maximin Approach

- Maximin approach is aka Pessimistic/Conservative approach
- It Maximizes the minimum payoff (Best of the Worst)

	Economy			
Alternatives	Growing	Stable	Declining	Worst
Bonds	40	45	5	5
Stocks	70	30	-13	-13
Mutual Funds	53	45	-5	-5

Decision: Invest in ***Bonds***

Minimax Regret and Laplace Rule

Minimax Regret

- Regret = opportunity loss
- **Regret = Best Payoff – Payoff Received**
- It Minimizes the Maximum regret (Minimum of Maximum)

Economy

Alternatives	Growing	Stable	Declining
Bonds	70 – 40 = 30 40	45 – 45 = 0 45	5 – 5 = 0 5
Stocks	70 – 70 = 0 70	45 – 30 = 15 30	5 – (-13) = 18 -13
Mutual Funds	70 – 53 = 17 53	45 – 45 = 0 45	5 – (-5) = 10 -5

Regret Table

Economy

Alternatives	Growing	Stable	Declining	Maximum
Bonds	30	0	0	30
Stocks	0	15	18	18
Mutual Funds	17	0	10	17

Laplace Rule

- Laplace approach is also know as Equally Likely approach
- It Maximizes the average payoff (Best of Averages)

Economy

Alternatives	Growing	Stable	Declining	Average
Bonds	40	45	5	(40+45+5)/3 = 30
Stocks	70	30	-13	(70+30-13)/3 = 29
Mutual Funds	53	45	-5	(53+45-5)/3 = 31

Decision: Invest in ***Mutual Funds***

Hurwicz approach

- Hurwicz approach is also know as **Realism approach**
- It finds the **compromise b/w the best & the worst payoffs**
- The best weighted average payoff is chosen based on **coefficient of realism** α alpha which has values from 0 to 1
  - When α alpha is close to 1, the decision maker is **optimistic**
  - When α alpha is close to 0, the decision maker is **pessimistic**

α (best payoff) + $(1 - \alpha)$(worst payoff) $\alpha = 0.3$ $1 - \alpha = 0.7$

Alternatives	Growing	Stable	Declining	Weighted Average
Bonds	40	45	5	.3(45)+.7(5) = 17.0
Stocks	70	30	-13	.3(70)+.7(-13) = 11.9
Mutual Funds	53	45	-5	.3(53)+.7(-5) = 12.4

Decision: Invest in ***Bonds***

Summary

- Goals, Estimates, and Plans
- Estimation Techniques
- Addressing Uncertainty
- Prioritization
- Decisions under Risk
- Decisions under Uncertainty

CSE405-SEE Assignment: *Practical Considerations*

- Given that there is no **economic friction** within the Software **Ecosystem**, ToyTimeInc., a mid-sized company that makes toys with multiple **offshore** branches **outsourced** the development of personnel database management system (PDBMS) to G-Tech. It is now time to decide the degree of security, usability, portability, robustness etc (**Good Enough**) to build into the system. Use the above scenario to discuss the following:

  Q1. How can the "Good Enough" Principle be applied in this case?

  Q2. What are the effects of Friction-Free Economy on ToyTimeInc.?

  Q3. How can ToyTimeInc. (consumer) add value to the consumed resources (PDBMS)?

  Q4. Why ToyTimeInc. Outsourced the development of PDBMS to G-Tech?

Instructions:

1. Due date: **25-06-2023**
2. Maximum of **2 students per group**
3. Write your Name and Registration Number Clearly
4. Send a digital copy to salisu.garba@slu.edu.ng
5. Plagiarism checker will be used and 0 marks will be awarded for plagiarized works

Q & A

Any Question?