APS111: Engineering Strategies & Practice I

Arnav Patil

Department of Electrical and Computer Engineering, University of Toronto

Preface and Acknowledgements

Engineering Strategies & Practice (ESP) is a two-course program within the first-year curriculum at University of Toronto's Faculty of Applied Science & Engineering. These are introductory design courses, meaning that students work in teams on projects brought forward by real-life clients. In the fall semester, all groups work on one of a few projects brought forward by the same client (Hart House in my year, and Sidney Smith Commons the year prior). In the winter semester, however, teams work on projects that come a wide variety of clients at UofT and in industry. The primary text for this course is *Designing Engineers: An Introductory Text*, written mostly by University of Toronto professors.

This notes package is in no way a complete representation of the course. That being said, these notes do a good job of covering the content that will be covered in the final exam.

This study guide was written in LaTeX using Overleaf. The source project can viewed here [click me!]. Additionally, my notes for APS111 (ESP I) can be found here [click me!]. If you find any errors (or just want to provide feedback) feel free to reach out to arnav.patil@mail.utoronto.ca!

Lecture 2-1

① Engineering Design Process has five main stages

↳ Design Brief

↳ Information Gathering (Researching)

↳ Problem Statement and Requirements

↳ Solution Generation

↳ Evaluation and Testing

• Idea is to take a client-provided design brief and convert it into a problem statement

Design Brief → Problem Statement

✓ • Describes design problem

✓ • Engineering perspectives

✓ • Solution Independent

② • Focus on finding and removing ERRORS, BIASES, and IMPLIED SOLUTIONS

③ • The Gap explains the current state of the world and what is missing

The World as it is today → How the client wants things to be in the future.

$\boxed{\text{The Gap}}$

④ • Analyze the client statement: Gather information

↳ Add: information through research and observation

• Areas of research will depend on the project → context

↳ Multiple modes of research always useful. → TRIANGULATE your findings

⑤ • Define the need → must be SOLUTION-INDEPENDENT

⑥ Engineering Design Project Requirements

• Problem Statements
↳ Intro to design project ← Information gathering through research and observation
↳ Briefly describes gap need, scope, and context
↳ Does not describe a solution

+

Detailed Requirements: Details FOCs that define the design problem including a detailed description of the context

Solution Independent

Lecture 2-2

① Problem Analysis → Need and Scope

② • 5 Whys method   Client Statement } Implied Solution

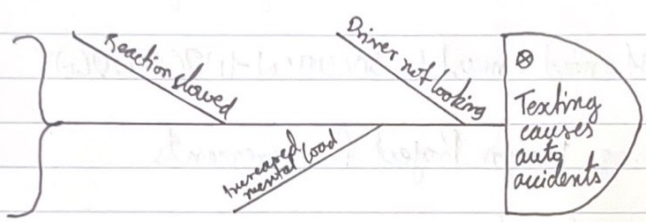
→ why → why → Out-of-sope

Need lies somewhere in here.

③ • Can also be used to find alternate scope.

⑤ Ishikawa diagrams → "fishbone diagram"

↳ A way of visualizing root cause - may help identify need, gap, or scope



⑥ Now we can define gap and need:

↳ Gap → Difference b/w current world and desired world. What is missing?

↳ Need → Solution-Independent description of the root problem

• Scope → What are dimensions of problem we want to solve

↳ How much of the gap is being addressed.

⑦ Project Triangle → can only control 2 out of 3

⑧ • How can the scope go wrong?

↳ Too big / narrow

↳ Wrong problem addressed

↳ Implied solution identified

↳ Uncontrolled change over time : scope creep

⑨ Problem Statement → Brings together essentials of gap, need, and scope and other key requirements

Lecture 2-3

② The Context : Service Environment

↳ Technology is designed for an operating environment

↳ Does NOT describe the design or how it will operate

↳ Uses ranges mainly but also averages

(8) Common pitfalls when writing Service Environment

↳ Interpreting effect of service environment rather than describing it.

↳ Failing to filter out irrelevant aspects of the environment.

(9) DETAILED REQUIREMENTS

• Turning information gathered from

↳ Client Statement ↳ Research ↳ Thinking, analyzing,

Into:

↳ Engineering specifications → Functions: MUST do → Use SHALL instead.

→ Objectives: SHOULD be

↳ Constraints: CANNOT be

(10) Primary Function → Secondary Functions

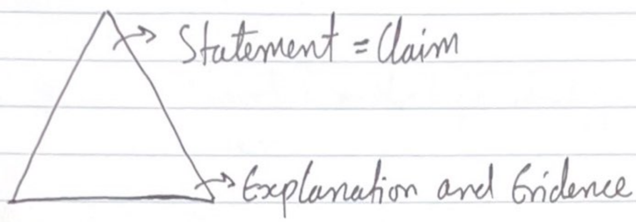
⇓

~~Means~~ Statements ⇒ Solution-Independent.

Lecture 3-1

• Investigation → Multimodal evidence gathering

• Documentation → An appropriate mix of modes to communicate relevant details



- Research → Primary → Evidence gathered by oneself
  - ↳ Secondary → Evidence gathered by others
- ↳ ☆Triangulation → Supporting claims with multiple sources of evidence

Lecture 3-3

- Inter-Team Communication Variables
- ↳ Synchronous / Real-Time
  - ↳ Message must be responded to before the next message is sent.
  - ↳ Sent within a set, dedicated time period.

☆ Necessary for conversation and decision-making

- ↳ Asynchronous / Time-Lagged
  - ↳ Messages sent independent of response

☆ Useful for updates and source items for discussion

Lecture 4-1

9 • Functional Basis → Powerful method for identifying functions

↳ All factors of functions come down to

↳ Energy ↳ Mass ↳ Information

10 • Pitfalls to avoid when writing functions

↳ Mix up functional basis with functions

↳ Writing means or results instead of functions

↳ Not using multiple tools to generate functions

• Methods used to generate functions

↳ Functional Basis ↳ Black Box Method

↳ Decomposition ↳ Benchmarking

11 • Black Box Method:

↳ Functions ↳ Objectives ↳ Constraints ↳ Solutions

Lecture 4-2

(5) • Objectives are part of the requirements

↳ What the design should BE!

OBJECTIVES → DESIGN REQUIREMENTS

CONSTRAINTS → DESIGN REQUIREMENTS

FUNCTIONS → DESIGN REQUIREMENTS

DESIGN REQUIREMENTS → GENERATE DESIGN

(6) • Objectives usually take form of adjectives

↳ Objectives broken up into goals that can be measured by metrics ⇒ not a Measure of Success

(9) Objectives → Goals → How-Why Chart

OBJECTIVE TREE	
Goals	Goals
Metrics	Metrics

HOW ↓ WHY ↑

(10) Objectives come from:

↳ Client statement and client communications

↳ Research: Publications, benchmarking, etc.

↳ Broader considerations: Service environment, stakeholders

(4) • Use Pairwise Comparison to Prioritize Objectives

↳ Pick the list of objectives at the same "level" when doing this prioritization

• Common pitfalls when writing objectives:

↳ Unmeasureable general objectives

↳ Making up goals without evidence

↳ Confuse objectives with constraints

↳ Confuse objectives with functions

Lecture 5-1

(5) • Constraints: What the design CANNOT BE.

(6) • Sources of Constraints → They are limits

↳ Clients ↳ Stakeholders ↳ Regulations, codes and standards

↳ Scientific Laws or Principles

(11) • If a design fails to meet an objective goal, then it is simply less desirable.

↳ However if it fails to meet a constraint, it is completely DISCARDED

13 • Common Pitfalls When Writing Constraints

↳ Creating unmeasureable constraints

↳ Creating unnecessary constraints

↳ Assuming all objectives have a corresponding constraint

Lecture 5-2

1

Mass IN	BLACK BOX	Mass OUT
Energy IN		Energy OUT
Information IN		Information OUT

5 • Iteration is important

PROBLEM STATEMENT → RESEARCH → FOC's → PROBLEM STATEMENT

Lecture 5-3

• Use the CRAAP method to evaluate sources:

↳ Currency → Relevance → Authority → Accuracy ⇒ PURPOSE

Lecture 6-2

• Project Requirements → a contract between a team and a client.

↳ Introduction / Problem Statement → Add precision and depth to client statement

↳ Broader Considerations → Stakeholders and service environment provide context.

↳ Detailed Requirements → Trustworthy measures to assess solutions

↳ All sections → Research, rigour, and a persuasive argument.

Lecture 6-3

① • Stakeholders → People or organizations who might:

↳ Be impacted by the design

↳ Have influence on the design

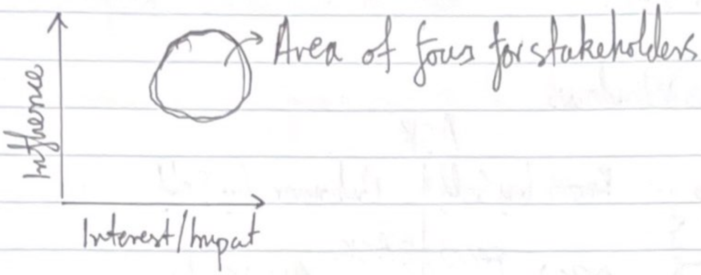
↳ Users and operators / clients not usually considered stakeholders

⑤ • Stakeholder Analysis

↳ Identify possible stakeholders ↳ Understand perspective

↳ Prioritize and sort the list ↳ Focus on key stakeholders

(8)



Lecture 7-2

(6) Decomposition - which type fits your project?

(7) • Functional Decomposition → Broken up by subfunction
↳ Shown with solution ideas

• Structural Decomposition → Broken up by components or subsystem

Lecture 8-2

(1) • Idea Generation

↳ Use multiple methods to achieve a wide variety of ideas

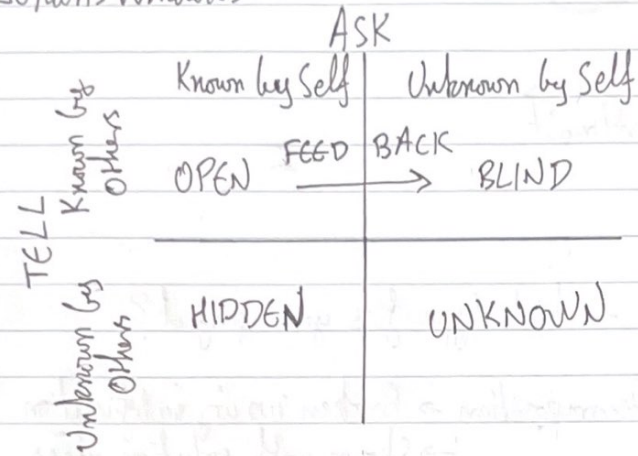
↳ Make sure you fully explore the design space, including "magic" ideas.

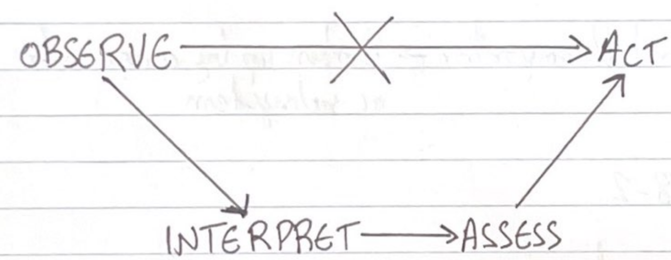
(3) • Structured Brainstorming → Individual then compile into a master list.

↳ ESP Recommended Way

Lecture 8-2.

• Johari's Windows





• Behavioral Feedback Model

↳ Action: Specific examples of behaviour

↳ Impact: How did this behaviour affect the team?

↳ Development / Desired Outcome: What is the desired change?

④ • Morphological Charting Method

↳ ① Decompose the problem → Use decomposition } Should be done by this point

↳ ② Use brainstorming or other methods for each part.

↳ ③ Combine them into an alternative solution.

⑦ • Apply the SCAMPER method to your solutions

↳ SUBSTITUTE

↳ COMBINE

↳ ADAPT

↳ MODIFY/MAGNIFY/MINIFY

↳ PUT to Other Use

↳ ELIMINATE

↳ REARRANGE /REVERSE

Lecture 9-1

• How to analyze team dynamics?

↳ Reflect on team organization

↳ Describe a situation

↳ Analyze situation using Team Charter

↳ Determine what change was made to the Team Charter as a result.

↳ Determine if the change was effective

- GRIP Model of Teamwork

↳ Goals: Everyone must be fully committed to the team

↳ Roles: All team members must know what role they play and how to be accountable

↳ Interpersonal: Quality communication and collaboration requires trust

↳ Processes: How teams solve problems and address conflict

Lecture 11-1

③ • 5 Steps to a Recommended Solution → Overview.

↳ ① Consolidate Team Idea List

↳ ② Feasibility Check

↳ ③ Multi-Voting Solutions

↳ ④ Graphical Decision Chart

↳ ⑤ Pugh Method

⑤ Step 1: Consolidate Team List

• Input → Multiple Individual Solution Lists

↓

• Merge everyone's ideas as a team → IDEA MASTER LIST

↓

• Briefly discuss each idea

↓

• Remove any duplicate ideas.

↓

• Output → One team list w/ 50-100+ solutions

⑥ Step 2: Feasibility Check

• Input: One consolidated master list of ideas.

↓

• Confirm each solution from master list Functions and Constraints

↓

• Output: If you cannot make it meet Figures and Constraints, discard

⑦ Step 3: Multi-Voting

- Input: List of 50+ feasible solutions

↓

- Output: Approximately 10 good solutions

⑧ Step 4: Graphical Decision Chart

- Input: Approximately 10 good solutions

↓

- Top-ranked objectives on x, second-top ranked on y

↓

- Through discussion score each design idea relative to the ranked Objective

↓

- Output: 3+ Great Alternative Solutions

⑨ Step 5: Pugh Method

- Input: 3 Great Alternative Solutions

↓

- Evaluate each alternative against a standard / reference solution

↓

- Output: 1 Design to recommend.

Lecture 12-1

① • What is the difference between a metric and MoS?

↳ Metric → An idea for how to measure an objective

↳ Measure of Success → A test procedure based on a metric to test the design

② • MoS: Bringing a metric for an objective into action

• Estimation → Quick calculation of an indicator of design

• Modelling → Numerical model to verify design

• Prototyping → One-off example of the design

⑦ • Prototyping should be as inexpensive and simple/fast to make as possible → "low-fidelity"

⑩ ACCURACY ≈ VALIDITY

ACCURACY ↓ Are you measuring the true value?

VALIDITY ↓ Are you measuring what you think?

⑪ PRECISION ↓ How close are my measurements?

RELIABILITY ↓ Is my measurement repeatable?

Lecture 12-3

(7) • Human-centred design or human centred systems design

• User-centred design - and the concept of affordance

• Interaction design → UI

• Experience design → UX

(4) • Human Tech Ladder ←

- PHYSICAL } ERGONOMICS
- PHYSIOLOGICAL } NORMS & PRACTICE
- SOCIAL
- ORGANIZATION } SOCIAL NORMS
- POLITICAL

Lecture 13-2

(8) • Intellectual Property

↳ Ideas are our scientific and business assets

• The law protects ideas in the form of:

↳ "Trademarks" → an identifier

↳ "Copyrights" → the right to copy / distribution

↳ "Patents" → protect products, concepts, and ideas

PERUSALL NOTES

Organizing Teams:

- We use the Tuckman Team Model
  - ↳ First three stages (forming, storming, norming) are for teams just starting out.
- Forming Stage
  - ↳ Team is coming together but still functions as a set of individuals
  - ↳ Dominated by teammembers thinking about themselves more than the team
- Storming Stage
  - ↳ Least comfortable stage of team lifespan
  - ↳ Teams can get stuck in this stage and may revert to individual decision-making and the project work returns to individual pieces.
- Norming Stage
  - ↳ Team members develop strategies for accomodating differences and actively engaging all teams members.

Tools for Organizing Teams

• Forming

↳ Create a Team Charter to lay out protocols and envision the team beliefs

↳ Use agendas and keep team minutes

• Dealing with Storming → Conflict Management.

↳ ① Separate people from problem

↳ ② Focus on interests, not positions

↳ ③ Develop solutions that will benefit everyone

↳ ④ Define effective performance measures

Management Strategies

• Task and People Problems

↳ One example of Project Management method is RACI

↳ Responsible, Accountable, Consulted, Informed

Producing

• Performing Stage of Tuckman Team Model

↳ When teams are in the Performing Stage, the focus is on the tasks at hand and achieving the goals of the project

MISCELLANEOUS TERMS/NOTES

- Affordance → Recognizing that the way people do things is generally not based on what is possible, but based on usage implied by design
  - ↳ Most obvious function of UI/UX design
- Biomimetics → Process of adapting solutions from the biological world.
- Bathtub Curve → Possibility of failure of a design over its lifespan
- DFX → Design for a specific consideration
- Fault-Tree Analysis → Means of calculating the probability of an undesirable outcome.
- Means Statement → Specific solution idea or a means of solving the design problem
- Opportunity Cost → Concept that one has to give up an opportunity to do one thing in order to achieve another
- Reversal Method → A creativity method whereby the direction or sequence of things is reversed to generate new ideas
- Strategy of Least Commitment → At any point in the design process the design space is only reduced when necessary.

- Team Collocation → Having the whole team working together at the same time e.g. sprints

- TRIZ → "Theory of Inventive Design" → Set of 40 inventive principles that appear over and over in the solution of inventive problems

- Structural Decomposition → Splitting up the presumed structure of the solution into sub-structures

- Functional Decomposition → Splitting up functions of a design into a set of sub-functions to analyze each separately