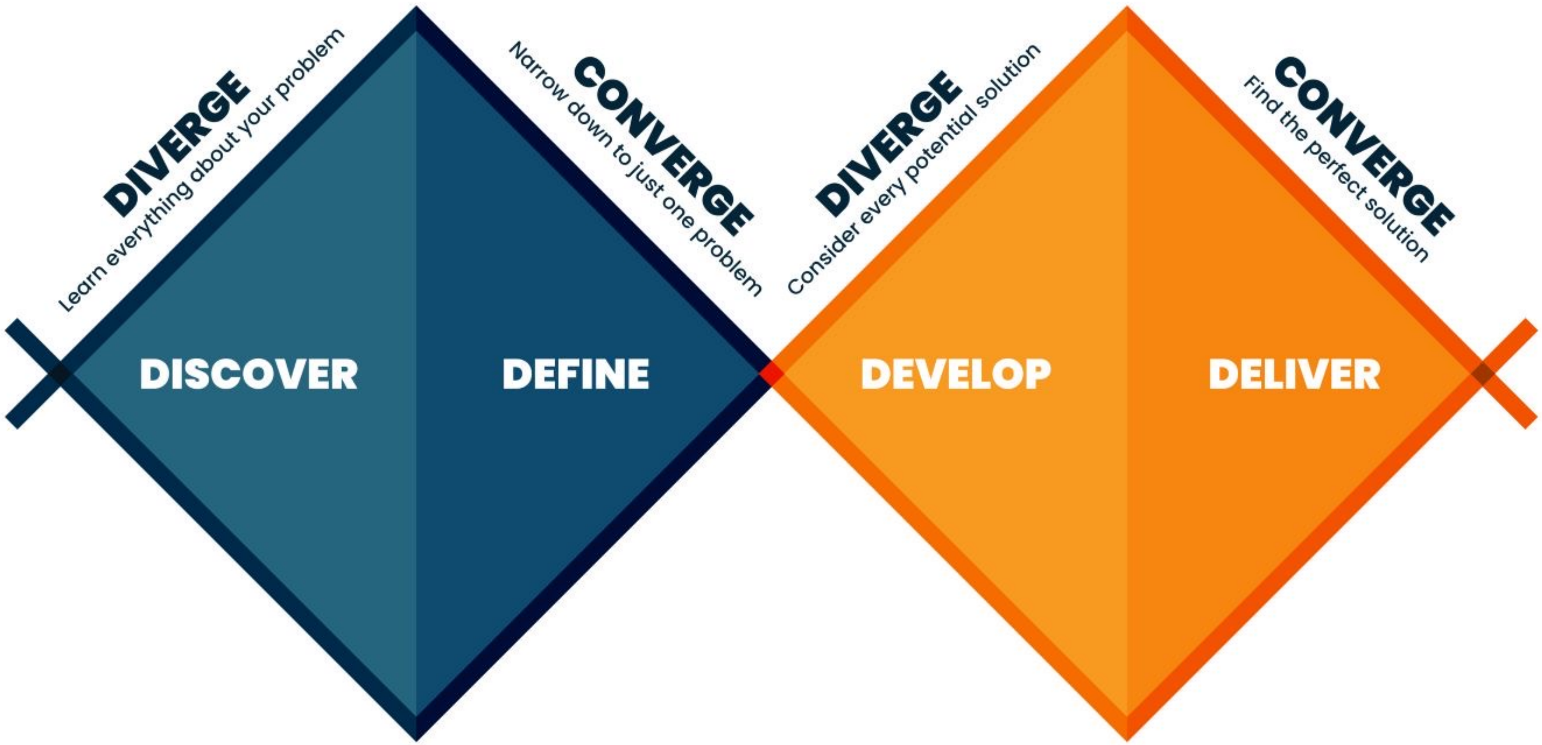
Phase 2 Define

DAEN 460 Capstone Senior Design

Alexander Abuabara

Spring 2026

ENGINEERING
TEXAS A&M UNIVERSITY


DIVERGE
Learn everything about your problem
CONVERGE
Narrow down to just one problem
DIVERGE
Consider every potential solution
CONVERGE
Find the perfect solution
DISCOVER
DEFINE
DEVELOP
DELIVER

- Critique your own project against **learning outcomes** rather than technical checklists only!
- You are not being graded on reproducing a solution but on the **quality of your thinking and decisions**!
- Peers are not parallel workers, but as part of a **shared system**. Shift the culture from individual performance to collective responsibility and make **collaboration** (not cooperation only)!
- **Metacognition** …
  why I'm thinking what I'm thinking → make the invisible visible!

Learning Outcomes *(read the syllabus!)*

Ability to:

1. **Identify**, **formulate**, and **solve** complex engineering problems by applying principles of engineering, science, and mathematics.
2. **Apply engineering design** to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors.
3. **Communicate effectively** with a range of audiences.
4. **Recognize ethical and professional responsibilities** in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts.
5. **Function effectively on a team** whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks and meet objectives.
6. **Develop and conduct appropriate experimentation, analyze and interpret data** and use engineering judgment to **draw conclusions**.
7. **Acquire and apply new knowledge as needed**, using appropriate learning strategies.

Shift From Engagement to Ownership

Unmotivated students often feel *done to*, not *in control*

1. Make Progress Visible (small steps)
2. Real Stakes (welcome to the real-world! bad models, biased algorithms, broken pipelines)
3. Reduce Cognitive Overload (break complex tasks into structured checkpoints)
4. Normalize Struggle Publicly (show common mistakes)
5. Increase Social Accountability (you are more likely to prepare when others depend on you!)
6. Introduce Constructive Friction (low challenge?)
7. **Use Structured Reflection** (what is still unclear? what felt right? what strategy helped you learn?)
8. Address the Hidden Variables (office hours, brief 1:1 check-ins)
9. Remember Our Standards and Support (high expectations/standards, we want Y'ALL to succeed!)

Today's Agenda

- **Virtual Project Showcase**
- **Communication Activity**
- **Customer Needs/Technical Objectives** (revisions/updates)
- **Requirements/Engineering Actions**
- **Systems Modeling**
- **ABET information**
- **Phase 2 Guidelines**

Virtual Project Showcase Registration

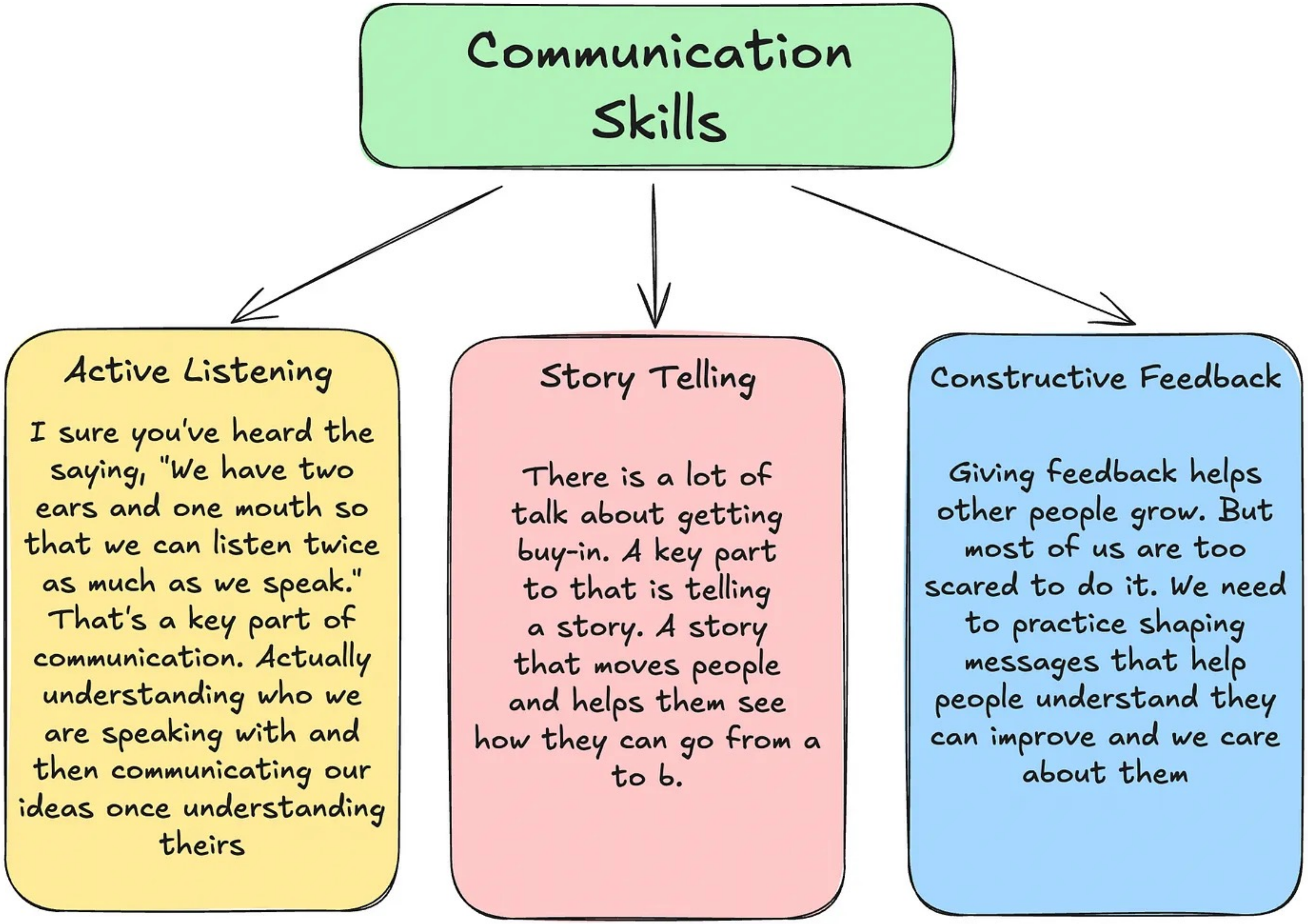
https://tamueps.secure-platform.com/students

Required assignment for all teams, submit proof on Canvas

Project title will identify your team in published material, ensure you get it correct!

Communication Activity

Communication is a Superpower



Ref.: [7 Lessons I Learned the Hard Way From 9+ Years as a Data Engineer]

Customer Needs
(*Voice of the Customer*)

- By now you should have a least <u>some</u> customer needs identified
  - Typically, this comes from:
    - The project description
    - Discussion with the sponsor/stakeholders
- Needs should not be directed towards a potential solution
  - Often a *flaw of needs* created without a thorough technical analysis

Customer Needs

- Garvin's dimensions of quality can provide the framework to identify needs
- As you model your "system", keep these important quality dimensions in mind to identify additional needs

Product Quality	**Dimension**	**Definition**
Product based approach	Performance	The primary operating characteristics of a product.
	Features	The secondary characteristics of a product that supplement its basic functioning.
	Durability	The product's probability of failure-free performance over a specified period of time.
Manufacturing based approach	Conformance	The degree to which a product's physical and performance characteristics meet design specifications.
	Reliability	A measure of useful product life, i.e., the amount of use a customer gets from a product before it deteriorates or must be replaced.
	Serviceability	The ease, speed, courtesy, and competence of repair.
User based approach	Aesthetics	How the product looks, feels, sounds, tastes or smells, a matter of personal preferences.
	Perceived	Quality based on image, brand name, or advertising rather than product attributes and, of course, is subjectively assessed.

Requirements
(Voice of the Engineer)

- Aka Engineering Characteristics
- Requirements are objective/testable/measurable
  - *Precise, clear, concise: "The system shall..."*
- Should address one or more (typically) of the customer needs
  - *Should be developed after a good representation and analysis of the problem!*
- Demonstrate how data engineering moves from *vague stakeholder expectations* ("fast, accurate, reliable data") to **clear, testable, operational requirements tied** to infrastructure, product characteristics, and process controls

Input / Process / Output (I/P/O)

Requirements Example

- I/P/O: *There exists a need for reliable, high-quality data to support near real-time business decisions*
- **Performance**
  - What does "near real-time" mean?
    - < 1 minute latency?
    - 5 minutes?
    - Hourly refresh?
- **Quality**
  - What constitutes "high-quality data"?
    - Completeness?
    - Accuracy?
    - Consistency across systems?
    - Schema stability?

Requirements Example Cont.

- I/P/O: *There exists a need for reliable, high-quality data to support near real-time business decisions*
- **Usability**
  - Can analysts easily access the data?
  - Is it queryable without specialized engineering support?
  - Are business definitions standardized?
- **Safety / Governance**
  - Is sensitive data protected?
  - Are regulatory requirements met (PII handling, auditability)?
- **Cost**
  - Comparable to existing cloud/data warehouse budgets
  - Competitive storage and compute utilization
  - Avoid unnecessary duplication of datasets

Requirements Example Cont. …

- **We generate a candidate set of <u>customer needs</u>:** statements, which if addressed by the "*product*" would resolve the I/P/O
- Interviews and investigation; leveraging Garvin's dimensions
- Requirements are then developed from the needs; precise, clear, concise, testable, measurable
  - *Data should be available within N minutes of source system update*
  - *Data should be at least 99% complete*
  - *Critical business metrics should be consistent across dashboards*
  - *Data pipelines should not fail silently*
  - *Users should not require engineering intervention to access curated datasets*
  - *Sensitive fields should not be exposed to unauthorized users*
  - *Data refresh schedules should be transparent and visible*

- Requirements are then developed from the needs: **precise, clear, concise, testable, measurable**
- Using the need as an organizing device for the requires:

  Data should be available within N minutes

  - **Product**
    - End-to-end pipeline latency ≤ 5 min (source commit to warehouse avail.)
    - 95th percentile latency ≤ 7 minutes
  - **Infrastructure**
    - Streaming ingestion supports ≥ 10,000 events/sec
    - Data warehouse auto-scales to maintain query latency ≤ 3 seconds for standard BI workloads
  - **Process**
    - Monitoring alerts triggered if latency exceeds SLA (Service Level Agreement) for > 2 consecutive intervals
    - Incident response time ≤ 15 minutes

- Requirements are then developed from the needs: **precise, clear, concise, testable, measurable**
- Using the need as an organizing device for the requires:

  Data should be accurate and complete

  - **Product**
    - Record completeness ≥ 99.5% per batch / Ref. integrity violations ≤ 0.1%
    - Schema validation enforced at ingestion
  - **Infrastructure**
    - Automated data quality checks executed on every load
    - Failed quality checks block downstream publication
  - **Process**
    - Daily reconciliation against source system totals
    - Root cause analysis completed within 1 business day for quality breaches

- Requirements are then developed from the needs: **precise, clear, concise, testable, measurable**
- Using the need as an organizing device for the requires:

  Data should be secure

  - **Product**
    - Role-based access control enforced at dataset and column level
    - PII fields encrypted at rest and in transit
  - **Infrastructure**
    - Access logs retained for ≥ 365 days
    - Audit events queryable within 24 hours
  - **Process**
    - Quarterly access review
    - Automated removal of inactive accounts after 30 days

- Requirements are then developed from the needs: **precise, clear, concise, testable, measurable**
- Using the need as an organizing device for the requires:

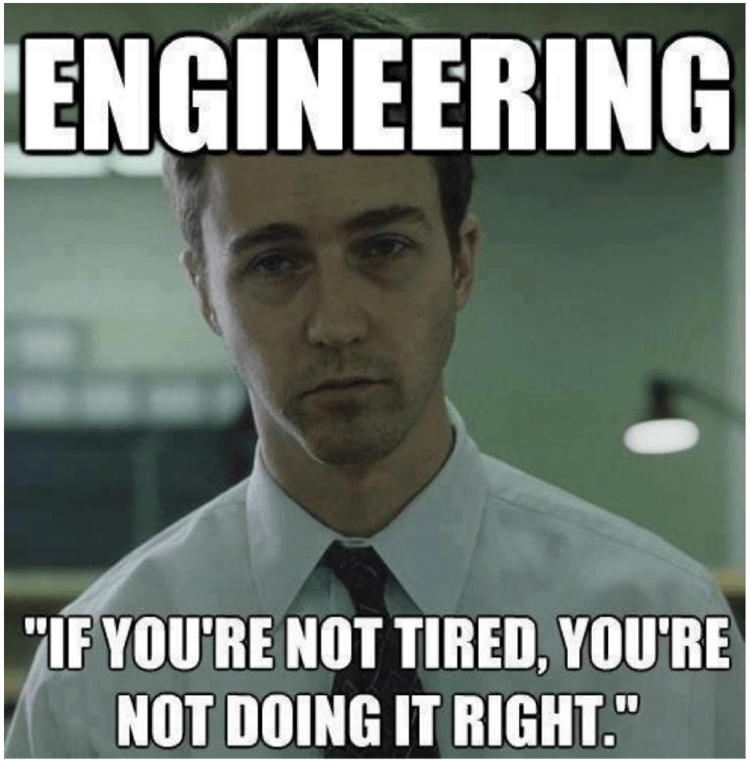
  Data platform should be cost-effective

  - **Product**
    - Storage cost per TB ≤ defined cloud budget threshold
    - Compute utilization ≥ 70% during peak workload windows
  - **Infrastructure**
    - Automatic suspension of idle clusters after 5 minutes
    - Lifecycle policies archive cold data after 180 days
  - **Process**
    - Monthly cost variance report delivered to stakeholders
    - Optimization review conducted quarterly

Engineering Actions

Recall "The Plan"

- This is where you start refining the plan
- What additional information is needed
- What needs to be done
    - By whom
    - By when
- What determines when the action is complete



Constraints

- Projects are subject to **constraints**
- They can be
  - Time
  - Financial
  - Resources (personnel and/or facilities)
  - Safety
  - Other

Note: The semester should not be listed as a constraint for your project. Specific time constraints related to your sponsor might be identified, but the semester schedule creates **deadlines**, not constraints.

Codes, Standards, and Specifications

Codes

- A code is a set of rules for performing some task, as in the local city building code or fire code.

Standards

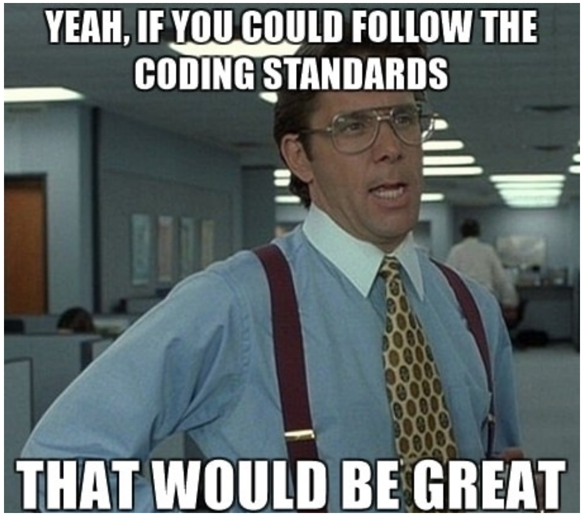
- A standard is less prescriptive and can be defined as a set of technical definitions and guidelines.

Specification

- A specification describes how a system should work, and is usually much more specific and detailed than a standard, but sometimes it is difficult to differentiate between documents that are called standards and those called specifications.

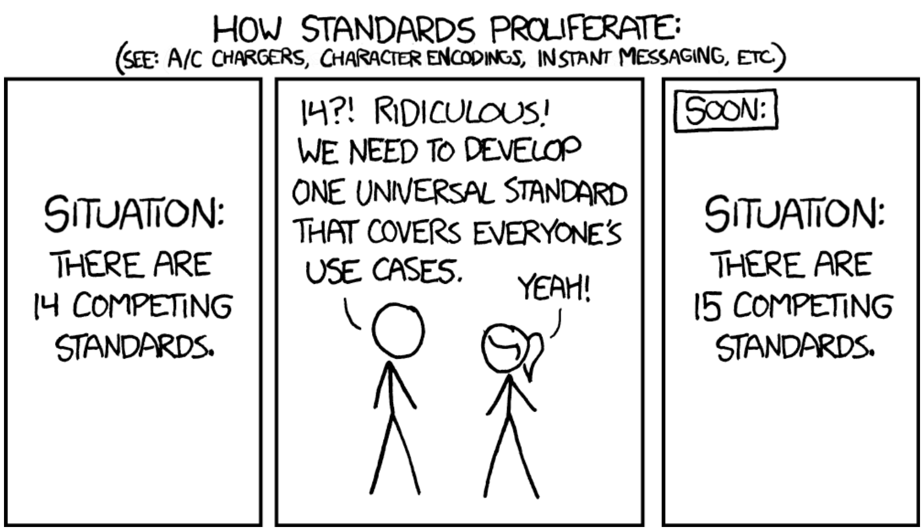
Where do you find applicable codes/standards?

- Your **project sponsor** should know specific codes/standards applicable to their systems and processes
- Your **faculty advisor** may be able to suggest codes that might apply
- **Engineering standards bodies** and organizations might have applicable codes



ANSI, NIST, ASTM, ASME, ISO

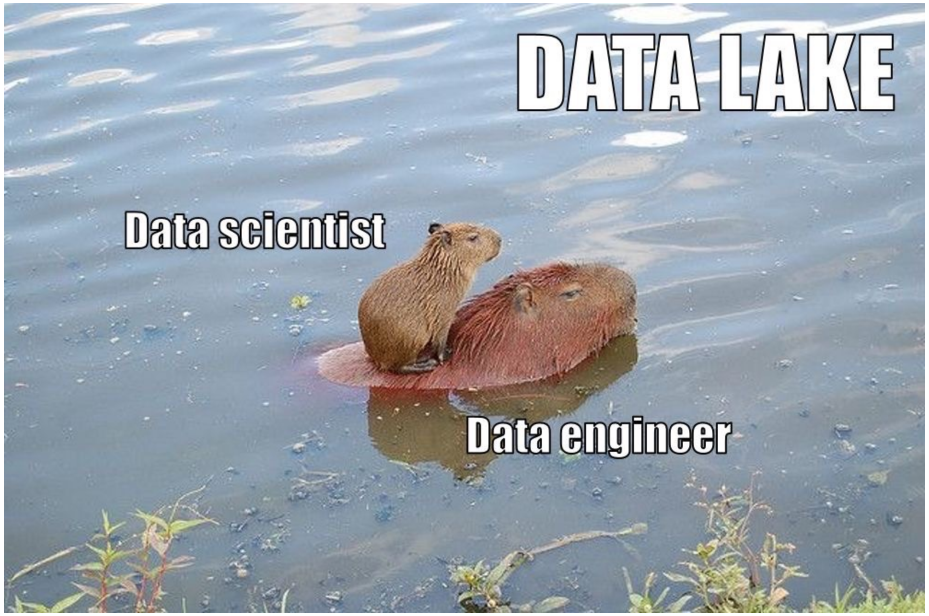
- The **American National Standards Institute** (ANSI) is the coordinating organization for the voluntary standards system for the United States (www.ansi.org)
- The standards responsibility of the U.S. government is carried out by the **National Institute for Standards and Technology** (NIST)
- **ASTM** International is the major organization that prepares standards in the field of materials and product systems
- The **ASME** prepares the well-known Boiler and Pressure Vessel Code that is incorporated into the laws of most states
- **ISO** is the international organization for standardization (www.iso.org)



Resources for Requirements and Actions

Resources are provided on Canvas to assist with requirements and actions:

- MIT Requirements Definition Presentation
- ISEN Requirements and Actions for Research Projects



Systems Modeling

- System modeling is the **process of creating abstract representations** of real-world systems to understand, analyze, design, or improve them.
- Instead of working directly with the full complexity of reality, we build **structured models** that capture the system's key components, relationships, inputs, outputs, and behaviors.
- A system can be **physical** (e.g., a manufacturing plant), **informational** (e.g., a database pipeline), **organizational** (e.g., a supply chain), or **computational** (e.g., a distributed software architecture).

Systems Modeling

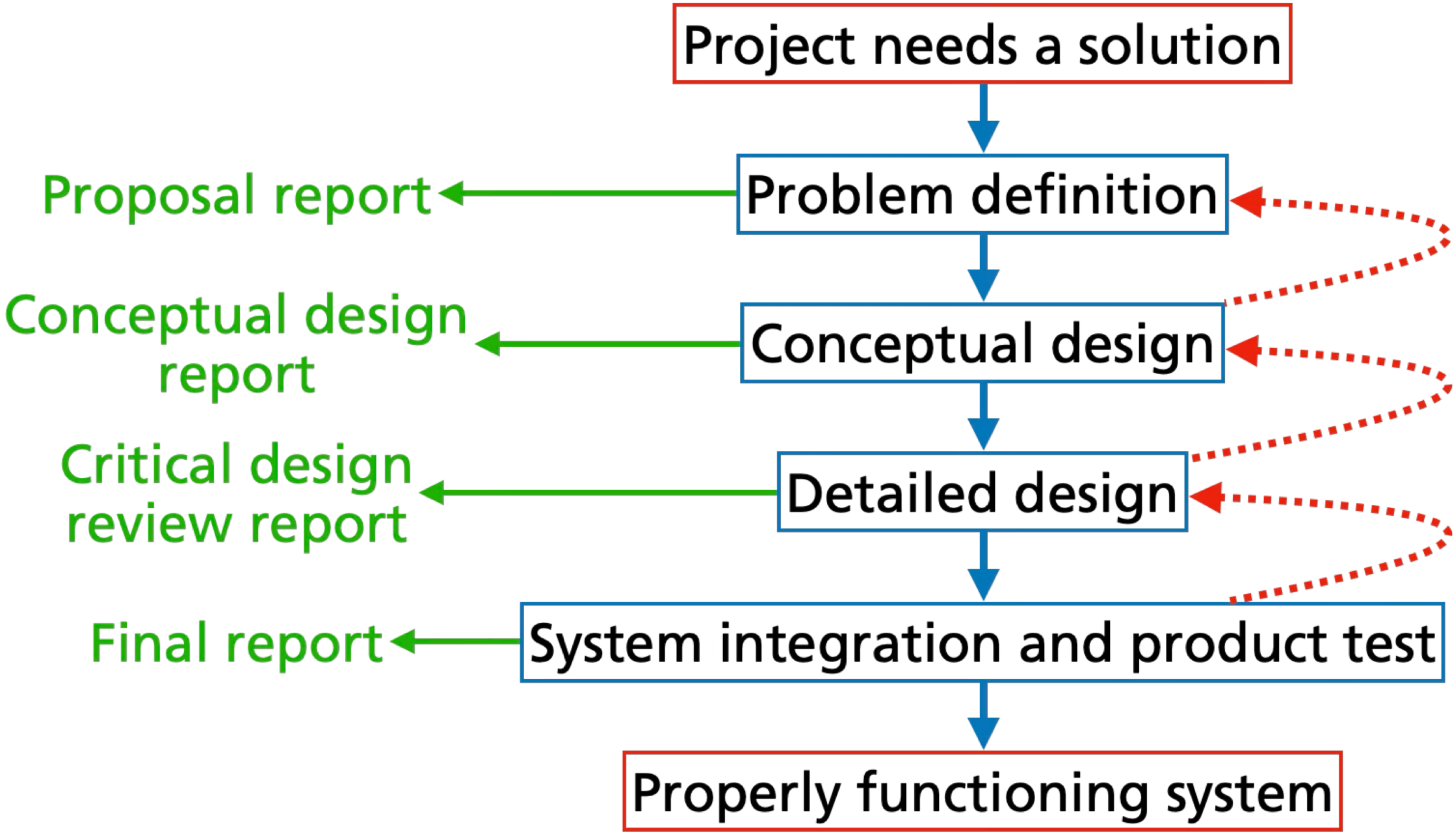
- At its core, system modeling is about **disciplined abstraction**: capturing what matters most so we can reason rigorously about complex systems!
- Common approaches include conceptual, mathematical, simulation, and data-driven models, each simplifying a system through **diagrams** (such as graphs or block diagrams), **mathematical equations** (such as differential equations), **simulations**, or **logical structures**.

Check examples: https://en.wikipedia.org/wiki/Systems_modeling

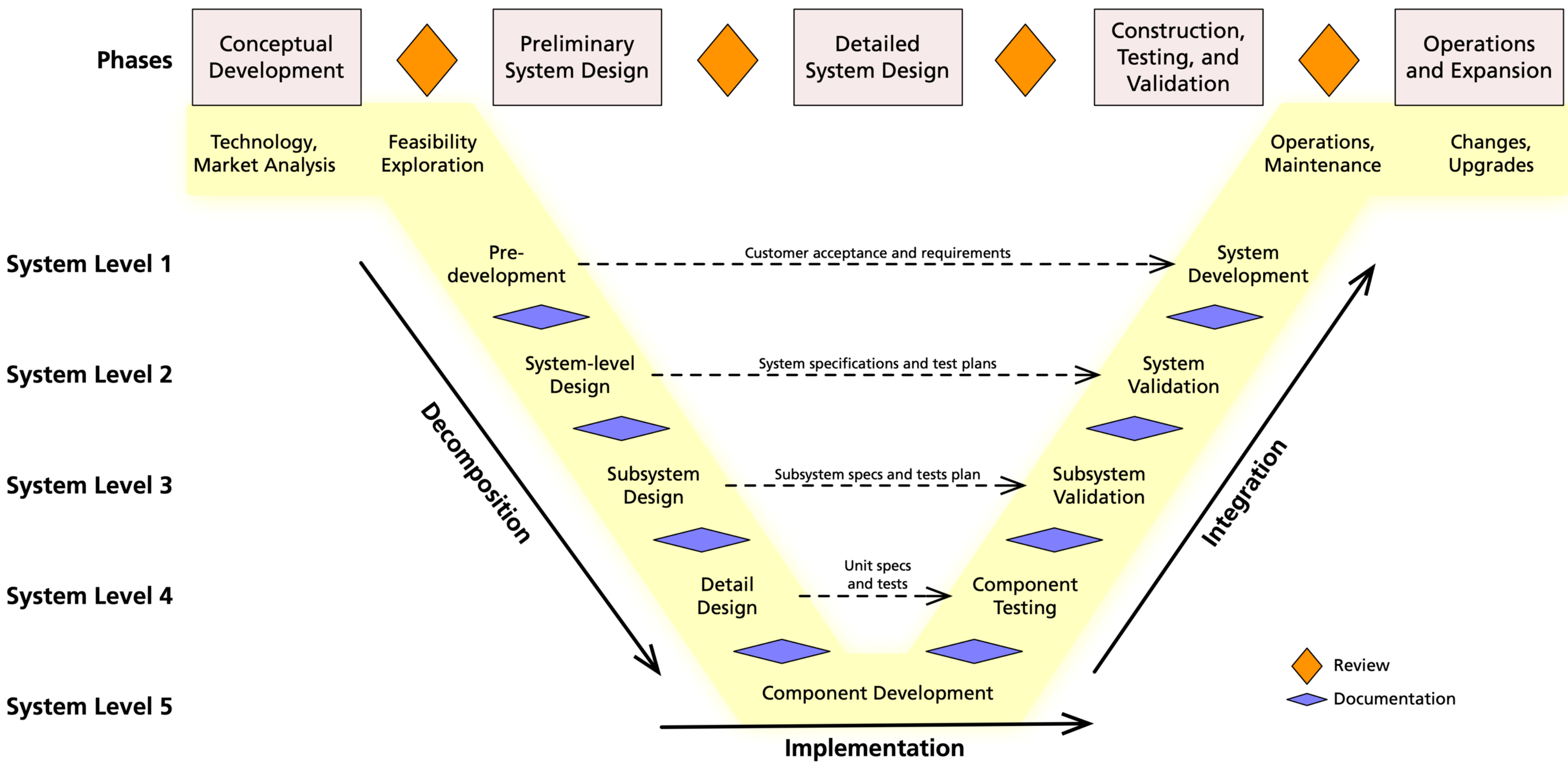
System Modeling helps to:

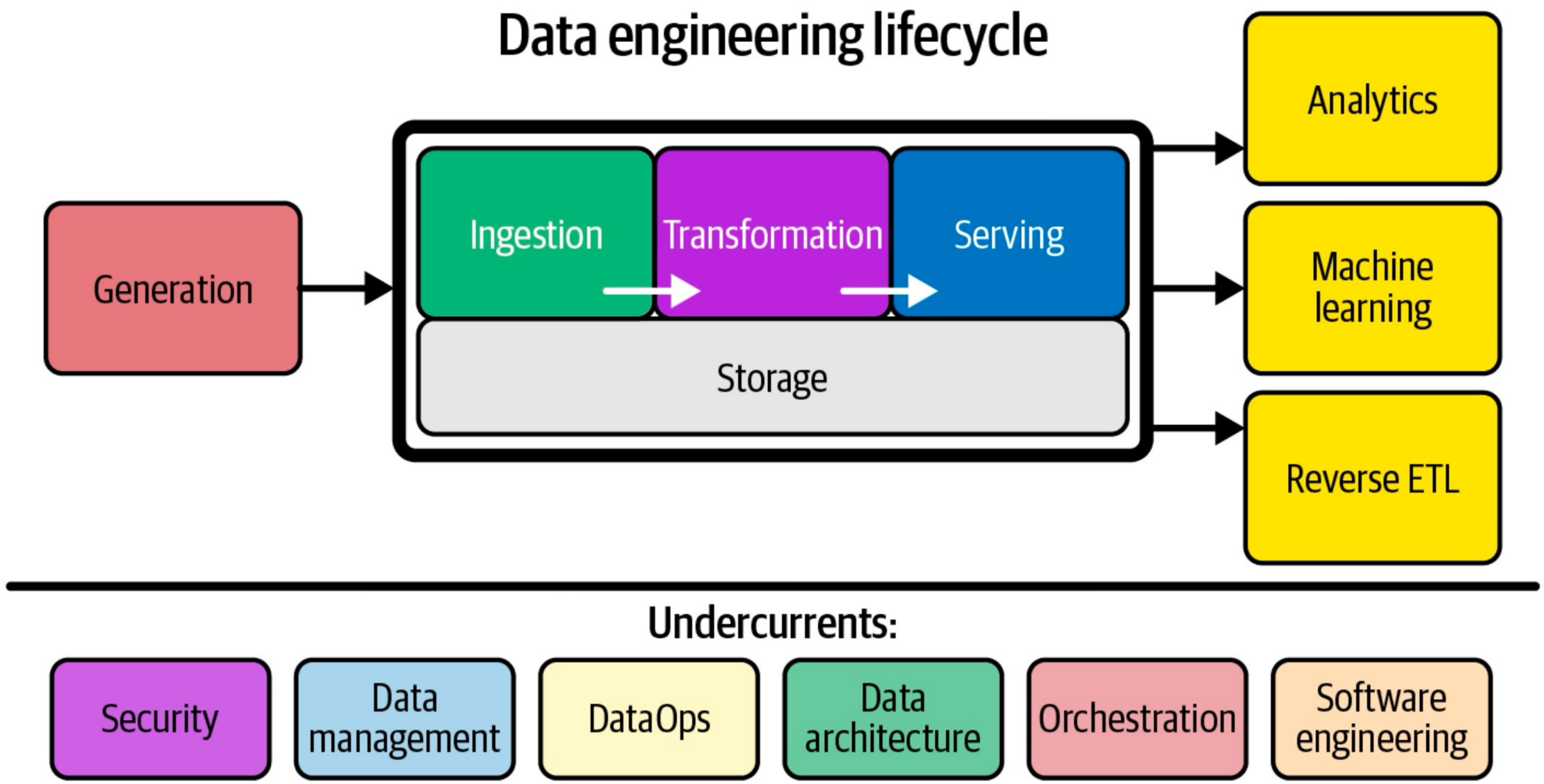
- Clarify how parts of a system interact
- Predict behavior under different conditions
- Identify bottlenecks or failure points
- Compare alternative designs before implementation
- Support optimization and decision-making

Design Process

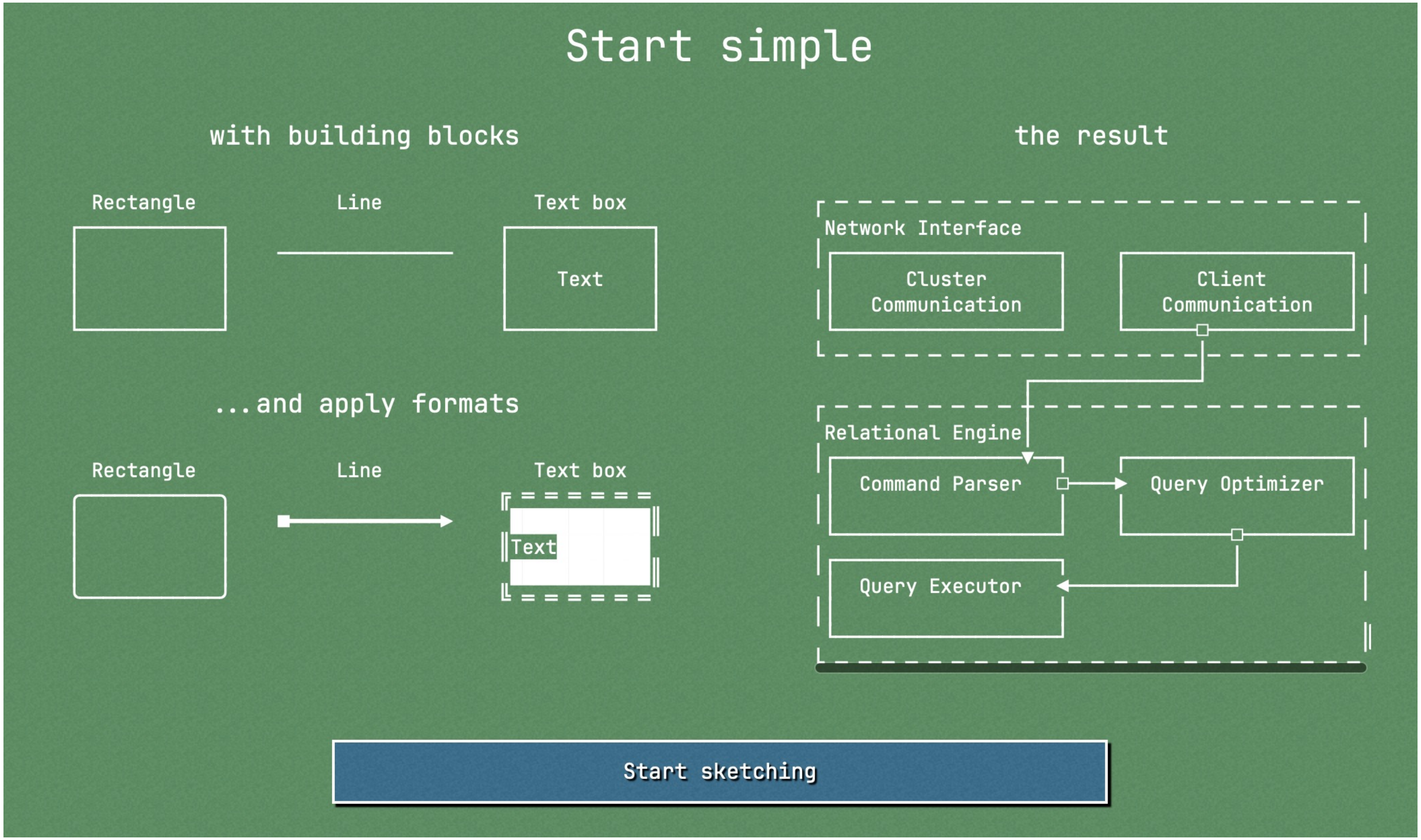


→ **Please review the slides from the 5th/6th week of last semester! link here**


Phases
Conceptual Development
Preliminary System Design
Detailed System Design
Construction, Testing, and Validation
Operations and Expansion
Technology, Market Analysis
Feasibility Exploration
Operations, Maintenance
Changes, Upgrades
System Level 1
System Level 2
System Level 3
System Level 4
System Level 5
Pre-development
Customer acceptance and requirements
System Development
System-level Design
System specifications and test plans
System Validation
Subsystem Design
Subsystem specs and tests plan
Subsystem Validation
Detail Design
Unit specs and tests
Component Testing
Component Development
Decomposition
Integration
Implementation
Review
Documentation


Data engineering lifecycle
Generation
Ingestion
Transformation
Serving
Storage
Analytics
Machine learning
Reverse ETL
Undercurrents:
Security
Data management
DataOps
Data architecture
Orchestration
Software engineering

MonoSketch



Start simple
with building blocks
Rectangle
Line
Text box
Text
...and apply formats
Rectangle
Line
Text box
Text
the result
Network Interface
Cluster Communication
Client Communication
Relational Engine
Command Parser
Query Optimizer
Query Executor
Start sketching

ABET Accreditation and Student Outcomes

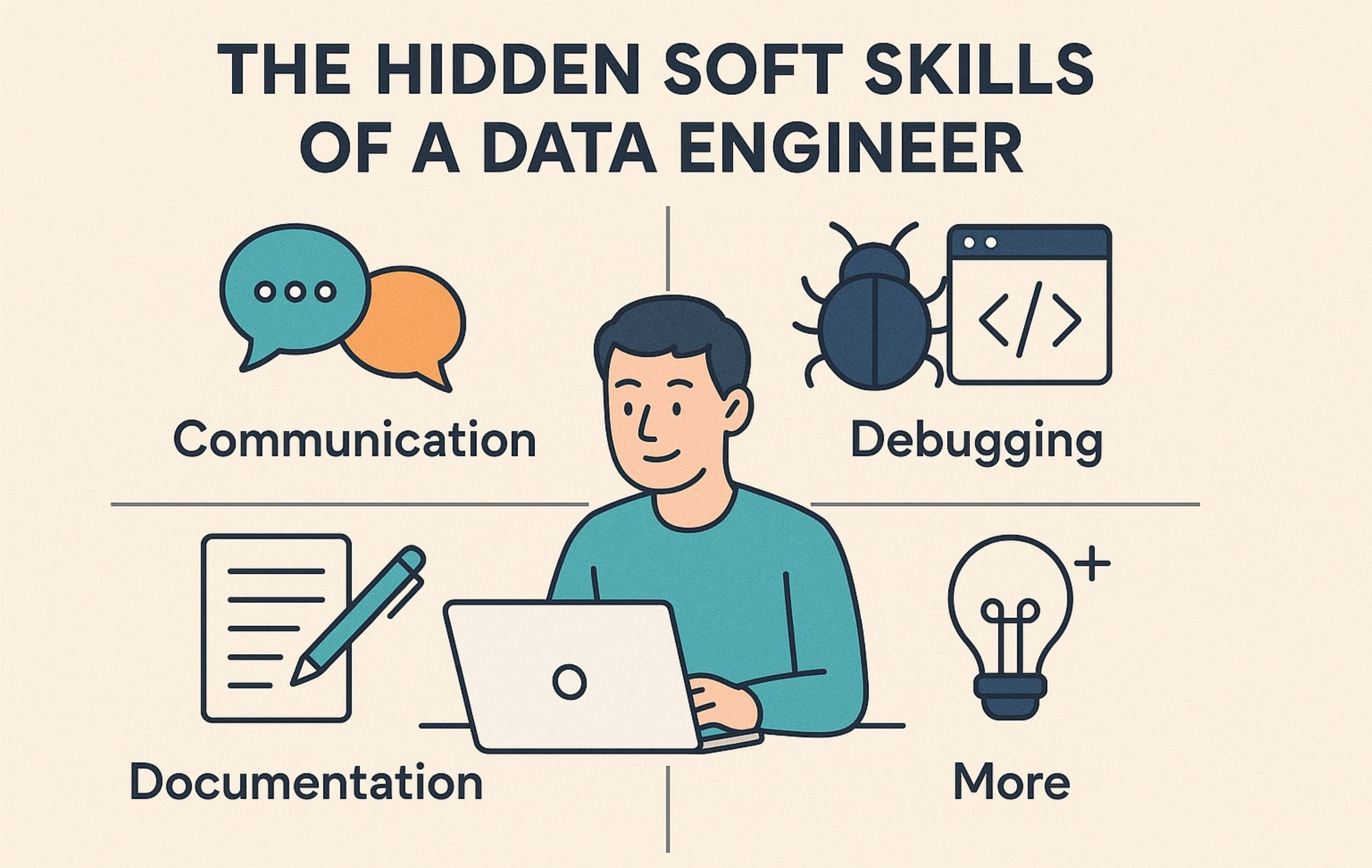
- ABET reviews the ISEN programs
- Demonstration of meeting established criteria must be documented
  - **Criterion 3, Student Outcomes**
    describes the expectations for students in and graduating
  - **Criterion 5, Curriculum**
    requires a culminating major engineering design experience that:
    - Incorporates appropriate **engineering standards** and **multiple constraints**
    - Is based on the **knowledge and skills** acquired in **earlier course work**
- **DAEN 460** is a primary course used for evaluation
- ABET is an important component of the value of your degree!

Documentation of Student Outcomes

- As part of your final report deliverables, your team will be expected to complete a questionnaire about student outcomes
→ ***SLIDE 4* in this presentation!**
- You will reference and link to sections of the report that meet the expected outcomes
- All outcomes need to be addressed
(*although some specific questions on the form may not be applicable*)

Phase 2 Guidelines

On Canvas



Nice reading: https://medium.com/@VijayRodrigues18/the-hidden-soft-skills-of-a-data-engineer-communication-debugging-documentation-and-more-25210d64e265