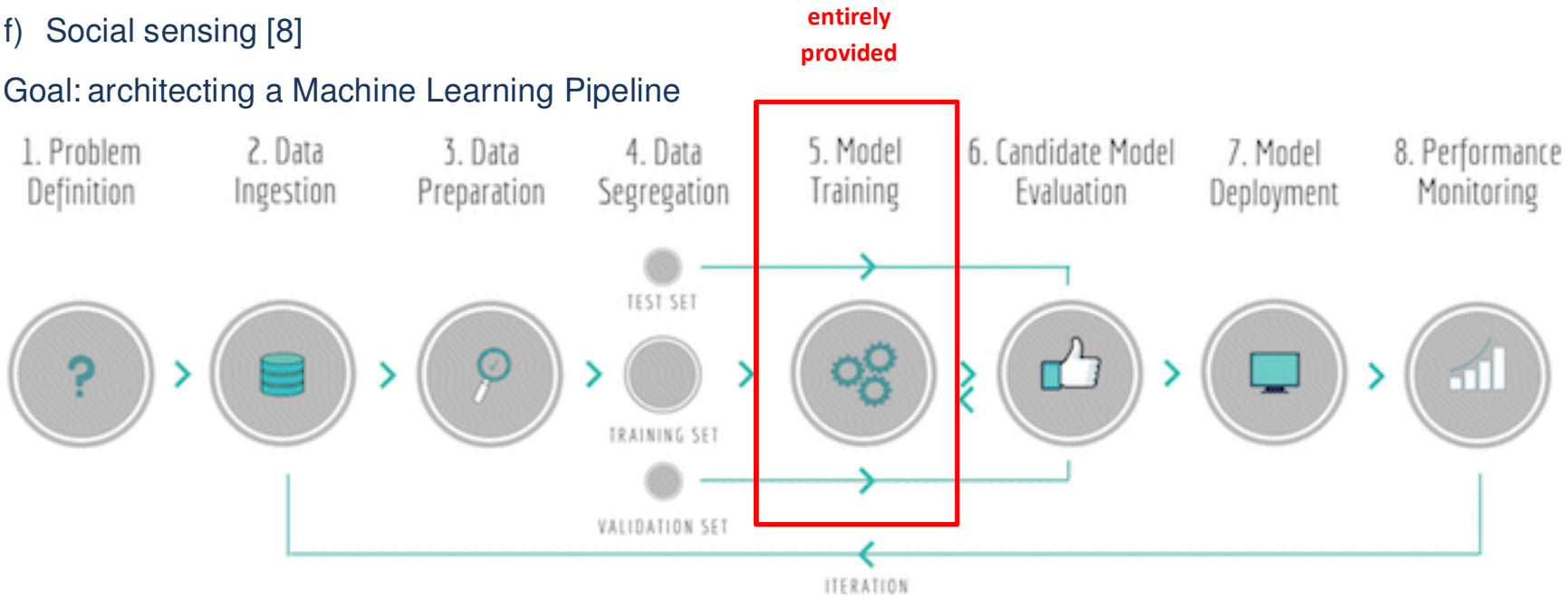
SSE - Introduction to Project Topics

- Team: 6 people
- Programming Language: Python
- Hours per member: 20 (in-class) + 60 (out-class)
- Topics [number of people]:
  a) Smart manufacturing (productivity) for curriculum “enterprise systems” [6]
  b) Smart manufacturing (quality) for curriculum “enterprise systems” [6]
  c) Adaptive smart robotics [6]
  d) Stigmergic memory [6]
  e) Smart shoes [6]
  f) Social sensing [8]
- Goal: architecting a Machine Learning Pipeline



1. Problem definition

a) Smart manufacturing (productivity)	**Input**: a time series periodically samples the number of products delivered by a machinery. **Output**: anomaly index in [0,1]; 1 anomalous series (e.g. overproduction, underproduction, intermittent production, peaks). **Model Training:** to learn the normal production with autoencoders **Training output:** to provide report about the learning quality
b) Smart manufacturing (quality)	**Input**: a time series periodically samples the number of products of a machinery that have been discarded. **Output**: anomaly index in [0,1]; 1 anomalous series (e.g. poor production, excellent production, intermittent quality). **Model Training:** to learn the normal quality with autoencoders **Training output:** to provide report about the learning quality
c) Adaptive swarm robotics	**Input**: a mission of target discovery in open environments made by a swarm of robots. **Output**: time needed to discover the 95% of targets; **Model Training:** to learn the best swarm parameters for each bio-inspired mechanism; **Training output:** to provide report about the learning quality
d) Stigmergic memory	**Input:** multivariate time series, e.g. EEG, wearable sensing, many application fields are possible

	Output: classification of the different pre-defined behaviors **Model Training:** to train the neural network for the lowest classification error; **Training output:** to provide report about the learning quality
e) Smart shoes	**Input:** a time series periodically samples pressure sensors **Output**: anomaly index in [0,1]; 1 anomalous series (e.g. long-term deviation from normal behavior). **Model Training:** to learn the normal and anomalous behavior via stigmergic perceptrons **Training output:** to provide report about the learning quality
f) Social sensing: encoding	**Input:** time series generated by terms occurrences in specific events discussed on Twitter **Output:** time series of events **Model Training:** to learn the dynamics and the categories with stigmergic perceptron **Training output:** to provide report about the learning quality
f) Social sensing: categorization	**Input:** time series of events generated by stigmergic perceptrons **Output:** clusters of terms **Model Training:** fuzzy clustering on relational data **Training output:** to provide report about the learning quality

2. **Data ingestion**

   Data collection

3. **Data preparation**

   Data exploration, data transformation and feature engineering.

4. **Data segregation**

   Split subsets of data to train the model and further validate how it performs against new data.

5. **Model training**

   Use the training subset of data to let the ML algorithm recognize the patterns in it.

6. **Candidate Model Evaluation**

   Assess the performance of the model using the test subset of data to understand how accurate the prediction is.

7. **Model Deployment**

   Once the chosen model is produced, it is typically deployed and embedded in decision-making frameworks.

8. **Model Scoring**

   Process of applying a ML model to a new dataset in order to uncover practical insights that will help solve a business problem. A.k.a. Model Serving

9. **Performance monitoring**

   The model is continuously monitored to observe how it behaved in the real world and calibrated accordingly.

Agile Project Management

- **Product Owner**: the person who represents the final user's best interest, and has the authority to say what goes into the final product.
- The Product Owner is in charge of making the **Backlog**, a list of tasks and requirements the final product needs. The backlog MUST be prioritized by the Product Owner.

- A **Sprint** is a predetermined **timeframe** within which the team completes sets of tasks from the Backlog. The length of time depends on the needs of the team, but two weeks is pretty typical.

- Teams meet periodically, according to their schedule, to give progress updates. Each Sprint ends with a **review**, or Retrospective, where the team reviews their work and discusses ways to improve the next Sprint.

- **Product Backlog:** The Backlog is where you list out everything the project needs, ordered by importance. Keep in mind that the Backlog is never complete. As the project takes shape and new needs emerge, you will add to this. The Product Owner takes primarily responsible for this.

- **Plan your Sprint:** Next, it is time to pick tasks from the backlog to be completed in your first Sprint. Sprints are time-limited. You can decide a time length that works for you, but they are always less than one month. During the Sprint Planning, the team decides what tasks to include in this Sprint and who will be responsible for them.

- **Get to work.** Time to start working on that Sprint. Team members work on their tasks, and everybody checks in on their progress at the Team Meeting. Very often, this meeting lasts no more than 15-minutes and answers three questions: What did you work so far? What will you work in the next future? Is there anything blocking your work today that you need help with?

- **Review your work:** At the end of the Sprint, the team reviews the work accomplished and presents their completed tasks.

- **Review your process:** During the Retrospective meeting, you'll review how the actual work process went and plan ways you can improve your work and be more efficient next time.

- **Repeat.** With your first Sprint complete, it's time to start over again. Pick more tasks from the Backlog and repeat the process.
- At each session of work, a member summarizes his progress in a **Journal of Activities**.

The Scrum Board

- The **Scrum Board** is a place where you can organize your Backlog, as well as tasks that are being worked on in the current sprint and their progress. Scrum Boards can be as simple as a whiteboard with sticky notes for each task, or as complicated as specialized software, with charts and task tracking features.
- Trello Scrum Board is broken up into seven lists (inspired by this blog post), representing the workflow of my tasks.

- **Resources:** In this list, I keep all tasks that are recurring. That way I don't have to make a new card every time I need to build a landing page for a webinar. Just move that card out from the Resources list.

- **Backlog:** Here's where I keep my Backlog of tasks to be worked on. When my boss tells me he has something he needs help with, I add it to my Backlog list.

- **To Do:** When I plan my Sprint, I pull tasks from the Backlog to this list. This is the current Sprint I'm working on.

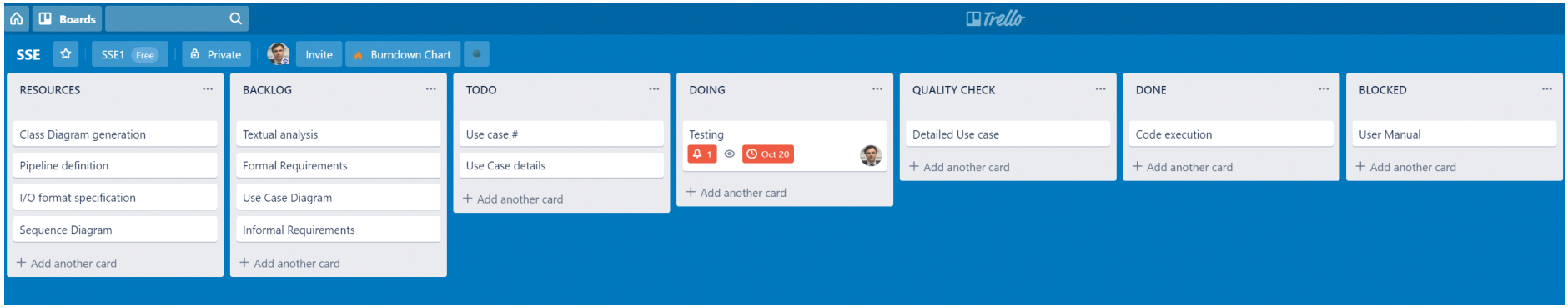
- **Doing:** When a task has been started, it gets moved here.

- **Quality Check:** As tasks are completed, they get moved to "QC." At the end of the week, I review this list to make sure everything is right.

- **Done:** Passed quality check, ready to be shipped. No more edits or reviews necessary, it's scheduled and ready for action.

- **Blocked:** When something is preventing me from completing a task (maybe I need to purchase something first and need approval from my boss), I move it to "Blocked", along with a comment about what the blocker is.

When a team member moves a task from "Doing" to "QC," I know they're ready to move onto the next task.


Boards
Trello
SSE
SSE1 Free
Private
Invite
Burndown Chart
RESOURCES
Class Diagram generation
Pipeline definition
I/O format specification
Sequence Diagram
Add another card
BACKLOG
Textual analysis
Formal Requirements
Use Case Diagram
Informal Requirements
Add another card
TODO
Use case #
Use Case details
Add another card
DOING
Testing
1
Oct 20
Add another card
QUALITY CHECK
Detailed Use case
Add another card
DONE
Code execution
Add another card
BLOCKED
User Manual
Add another card