Requirements and ML

Overview

- ML or data intensive systems vs traditional systems
- SW 2.0
- Requirements For ML
- ML For Requirements

Software intensive systems

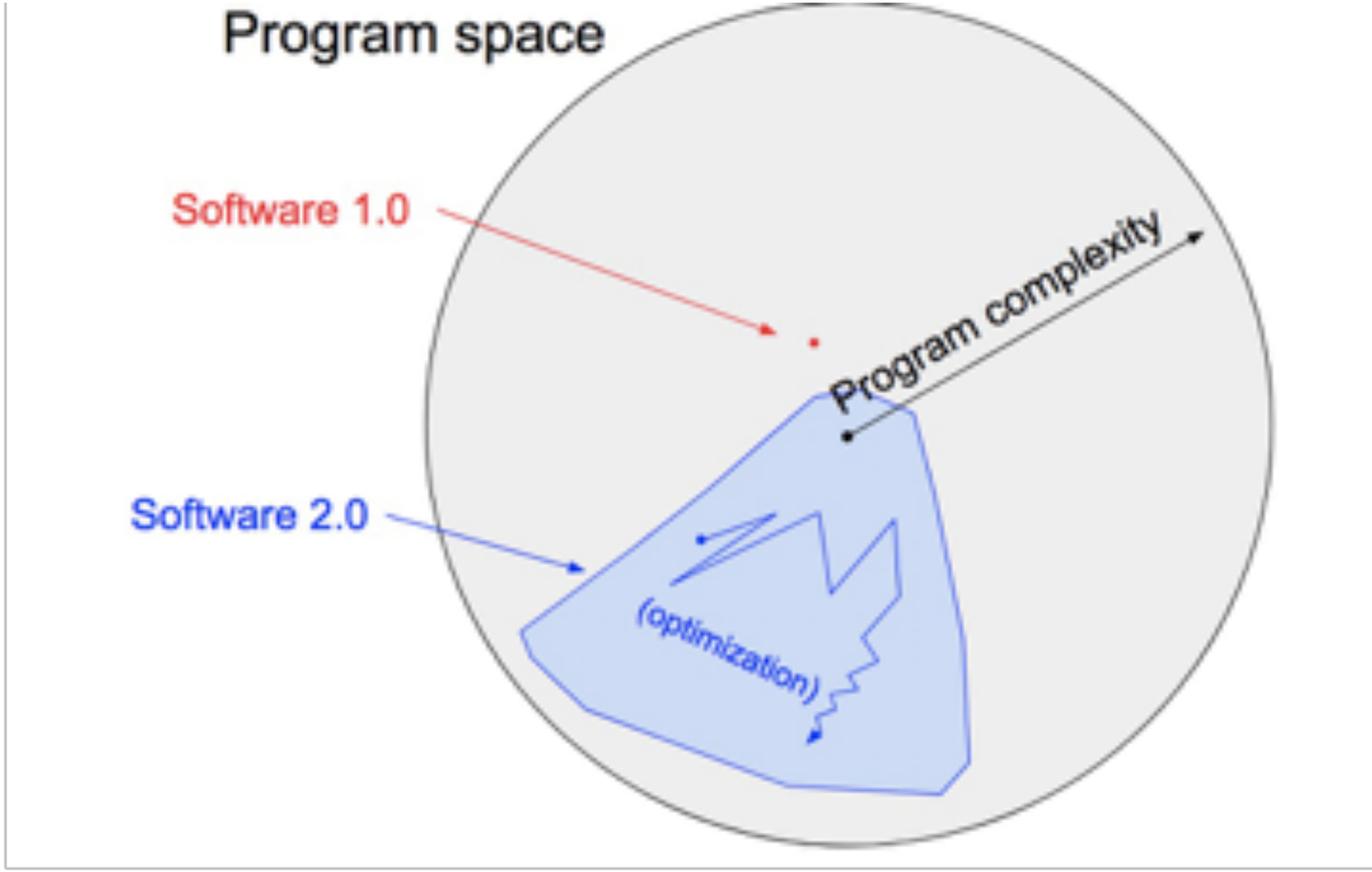
- Consider a robot. How do you tell the robot to get from A to B?
- Symbolic way: construct a model of the world. Program the robot with if/else rules to do path following.
- Connectionist/neural way: run the robot in many different paths and let it “learn” how to follow a path.

Software 2.0

- Andrej Karpathy
- Tesla windshield wiper system
- How does a Tesla know to not hit a car?
- When does a Tesla do poorly?
- How do you hack a self-driving car?

Software 2.0

BotSE



*We are the edge of a revolution in **how** we build software and **who** can "write" software"*

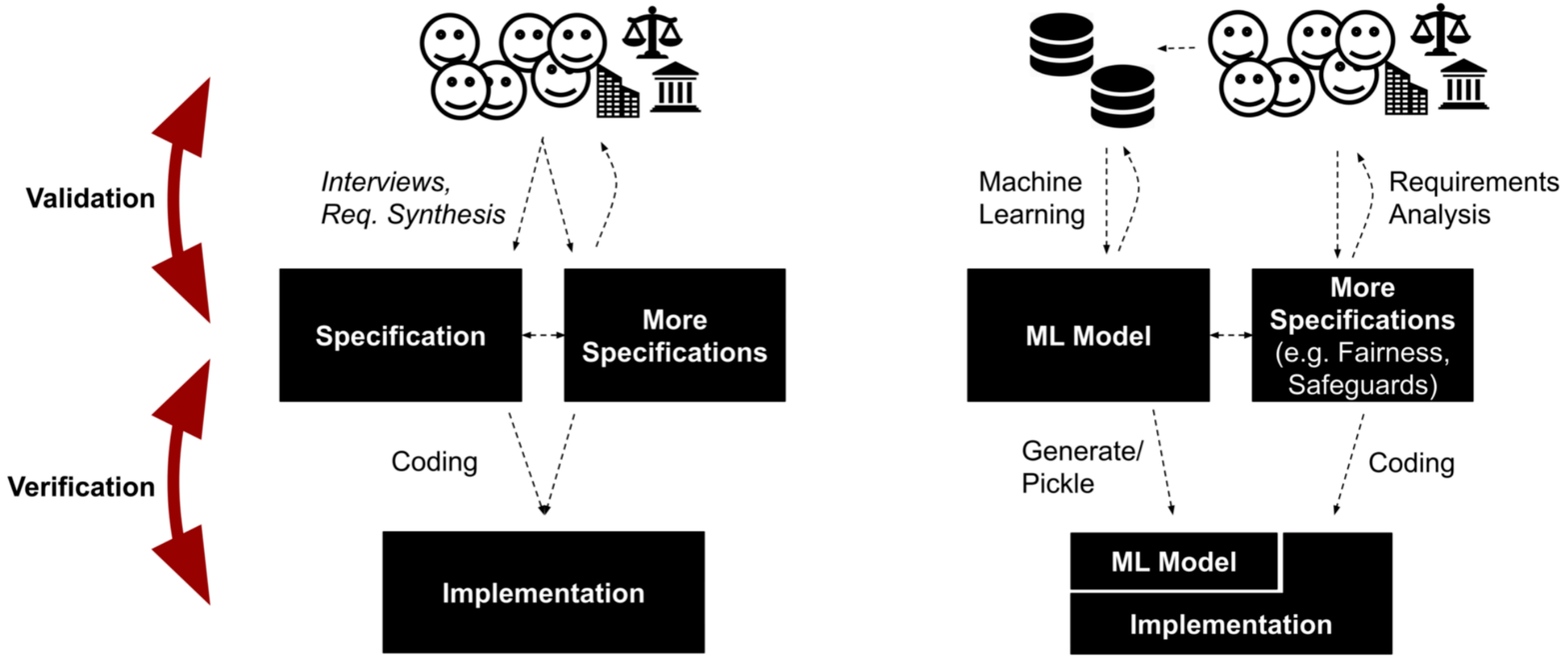
https://www.oreilly.com/radar/the-road-to-software-2-0/

Requirements for ML (RE4ML)

- What are requirements for a ML-based/connectionist system?
  - Are they different than conventional systems?
- Maybe ML is really just RE?
  - COMPAS: predict recidivism
    - Spec: IF age between 18–20 and sex is male THEN predict arrest (within 2 years)
    - ELSE IF age between 21–23 and 2–3 prior offenses THEN predict arrest
    - ELSE IF more than three priors THEN predict arrest
    - ELSE predict no arrest

RE4ML

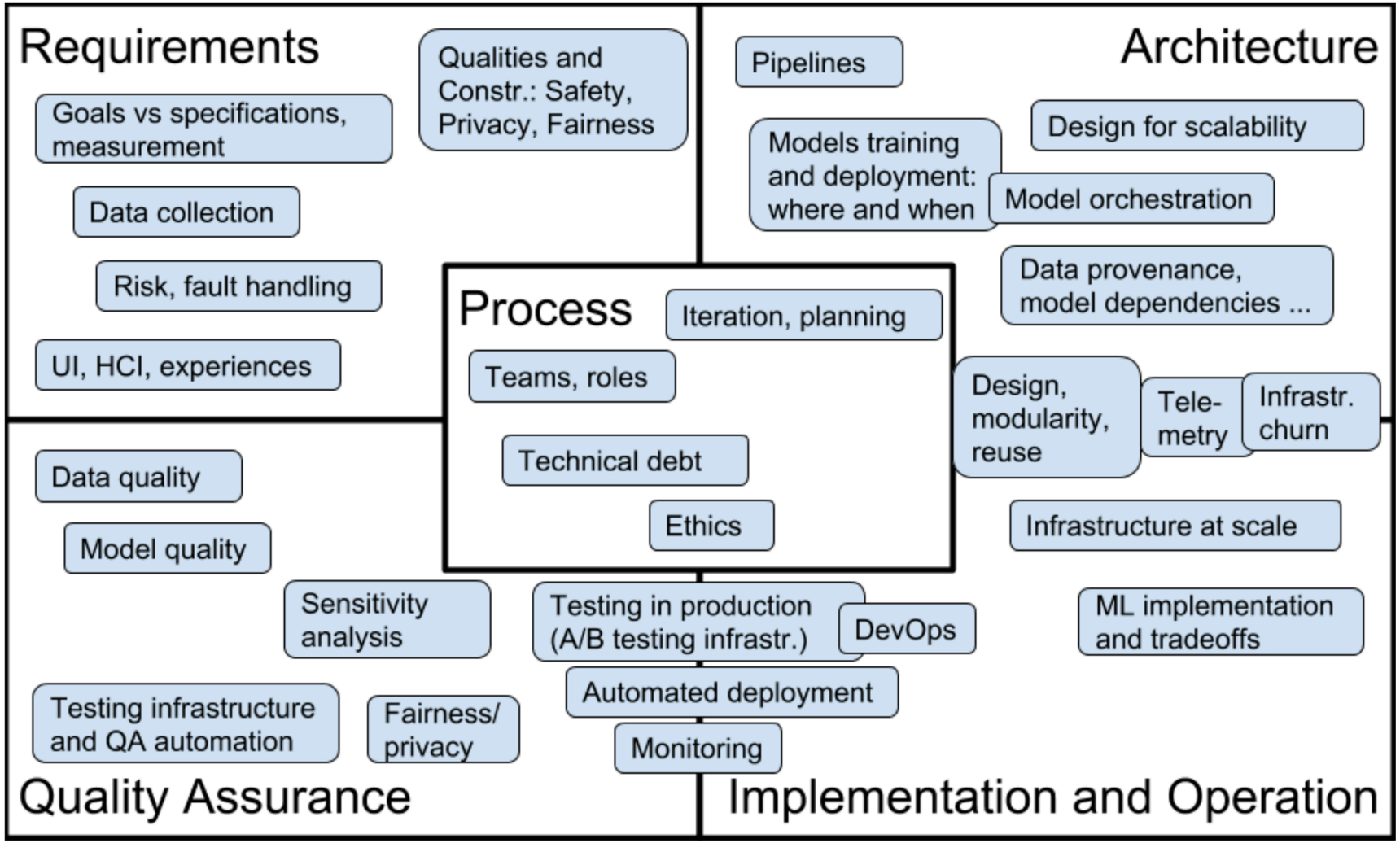
- So maybe a ML model is just a “learned” specification for the system?



Machine Learning as Requirements Engineering

https://medium.com/analytics-vidhya/machine-learning-is-requirements-engineering-8957aee55ef4

ML and Software Engineering



From Christian Kastner https://ckaestne.github.io/seai/S2020/#course-content

ML4RE

- We can leverage ML to help do requirements
  - Use NLP to find new requirements or detect problems
  - Use neural networks to find new requirements
  - Use SSBSE to find optimal configurations

NLP for requirements gathering

- Consider App Store reviews (show App Store)
- We can download these (show JSON)
- What should we do with this rich dataset?
- https://mast.informatik.uni-hamburg.de/app-review-analysis/

MARC - LSU

- https://github.com/seelprojects/MARC-3.0

NLP for Quality Checks

- As you've seen, requirements are non-trivial to write properly.
  - Ambiguity, vagueness, inconsistency, etc.
- Two approaches:
  - **Symbolic**: detect violations of structured NL
  - **NLP**: looks for ambiguity or other violations based on syntax analysis
- E.g., AQUSA tool (https://github.com/RELabUU/aqusa-core)
  - Lucassen, G., Dalpiaz, F., van der Werf, J.M.E.M. et al. Improving agile requirements: the Quality User Story framework and tool. Requirements Eng 21, 383–403 (2016). https://doi.org/10.1007/s00766-016-0250-x

Table 1 Quality User Story framework that defines 13 criteria for user story quality: details

Criteria	Description	Individual/set
Syntactic		
Well-formed	A user story includes at least a role and a means	Individual
Atomic	A user story expresses a requirement for exactly one feature	Individual
Minimal	A user story contains nothing more than role, means, and ends	Individual
Semantic		
Conceptually sound	The means expresses a feature and the ends expresses a rationale	Individual
Problem-oriented	A user story only specifies the problem, not the solution to it	Individual
Unambiguous	A user story avoids terms or abstractions that lead to multiple interpretations	Individual
Conflict-free	A user story should not be inconsistent with any other user story	Set
Pragmatic		
Full sentence	A user story is a well-formed full sentence	Individual
Estimatable	A story does not denote a coarse-grained requirement that is difficult to plan and prioritize	Individual
Unique	Every user story is unique, duplicates are avoided	Set
Uniform	All user stories in a specification employ the same template	Set
Independent	The user story is self-contained and has no inherent dependencies on other stories	Set
Complete	Implementing a set of user stories creates a feature-complete application, no steps are missing	Set

Table 2 Sample user stories that breach quality criteria from two real-world cases

ID	Description	Violated qualities
US_1	I want to see an error when I cannot see recommendations after I upload an article	*Well-formed* the role is missing
US_2	As a User, I am able to click a particular location from the map and thereby perform a search of landmarks associated with that latitude longitude combination	*Atomic* two stories in one
US_3	As a care professional, I want to see the registered hours of this week (split into products and activities). See: Mockup from Alice NOTE—first create the overview screen—then add validations	*Minimal* there is an additional note about the mockup
US_4	As a User, I want to open the interactive map, so that I can see the location of landmarks	*Conceptually sound* the end is a reference to another story
US_5	As a care professional I want to save a reimbursement—add save button on top right (never grayed out)	*Problem-oriented* Hints at the solution
US_6	As a User, I am able to edit the content that I added to a person's profile page	*Unambiguous* what is content?
US_7	As a User, I am able to edit any landmark	*Conflict-free* US_7 refers to any landmark, while US_8 only to those that user has added
US_8	As a User, I am able to delete only the landmarks that I added	
US_9	Server configuration	*Well-formed, full sentence*
US_{10}	As a care professional I want to see my route list for next/future days, so that I can prepare myself (for example I can see at what time I should start traveling)	*Estimatable* it is unclear what see my route list implies
EP_A	As a Visitor, I am able to see a list of news items, so that I stay up to date	*Unique* the same requirement is both in epic EP_A and in story
US_{11}	As a Visitor, I am able to see a list of news items, so that I stay up to date	US_{11}
US_{12}	As an Administrator, I receive an email notification when a new user is registered	*Uniform* deviates from the template, no "wish" in the means
US_{13}	As an Administrator, I am able to add a new person to the database	*Independent* viewing relies on first adding a person to the database
US_{14}	As a Visitor, I am able to view a person's profile	

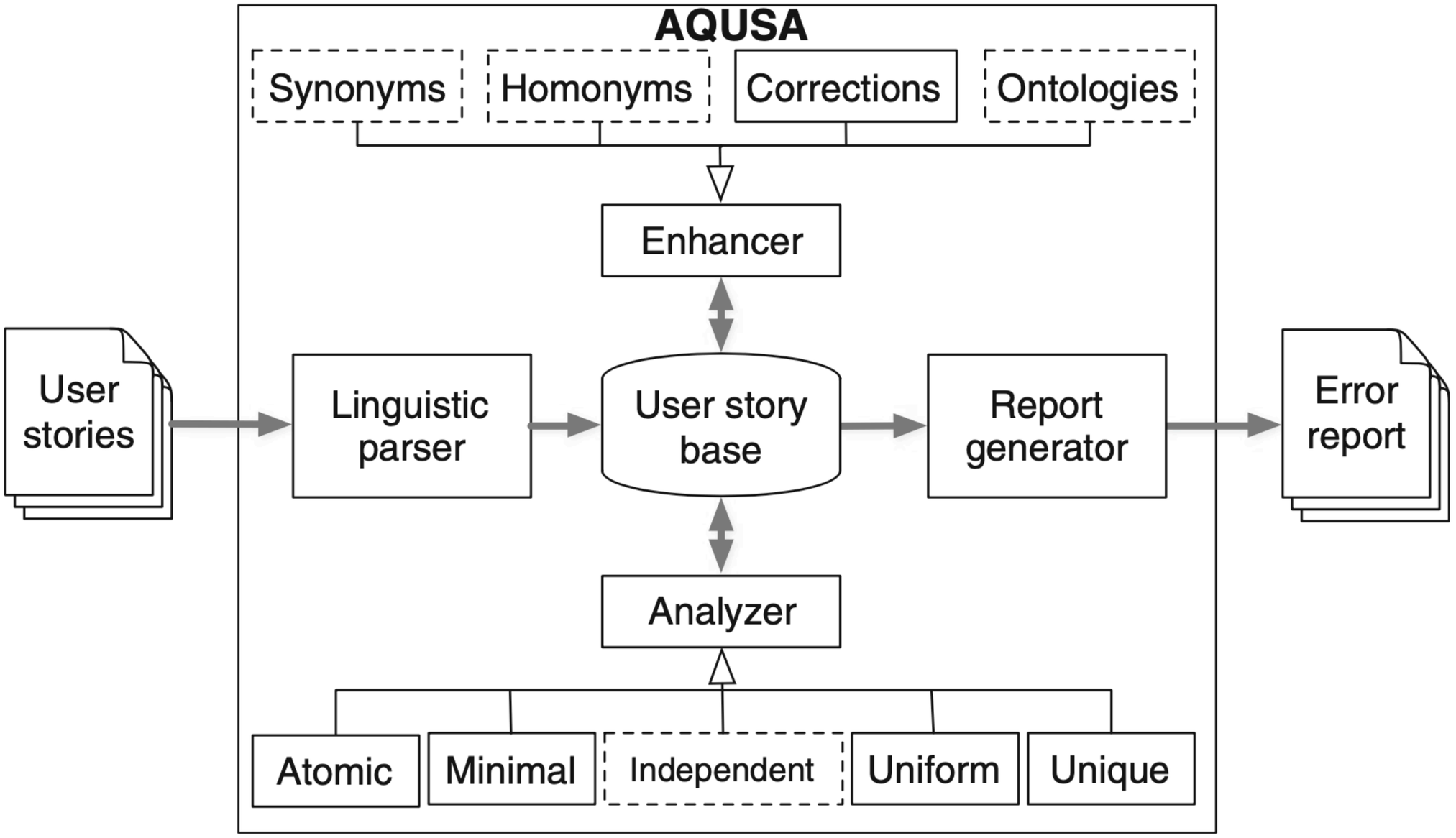
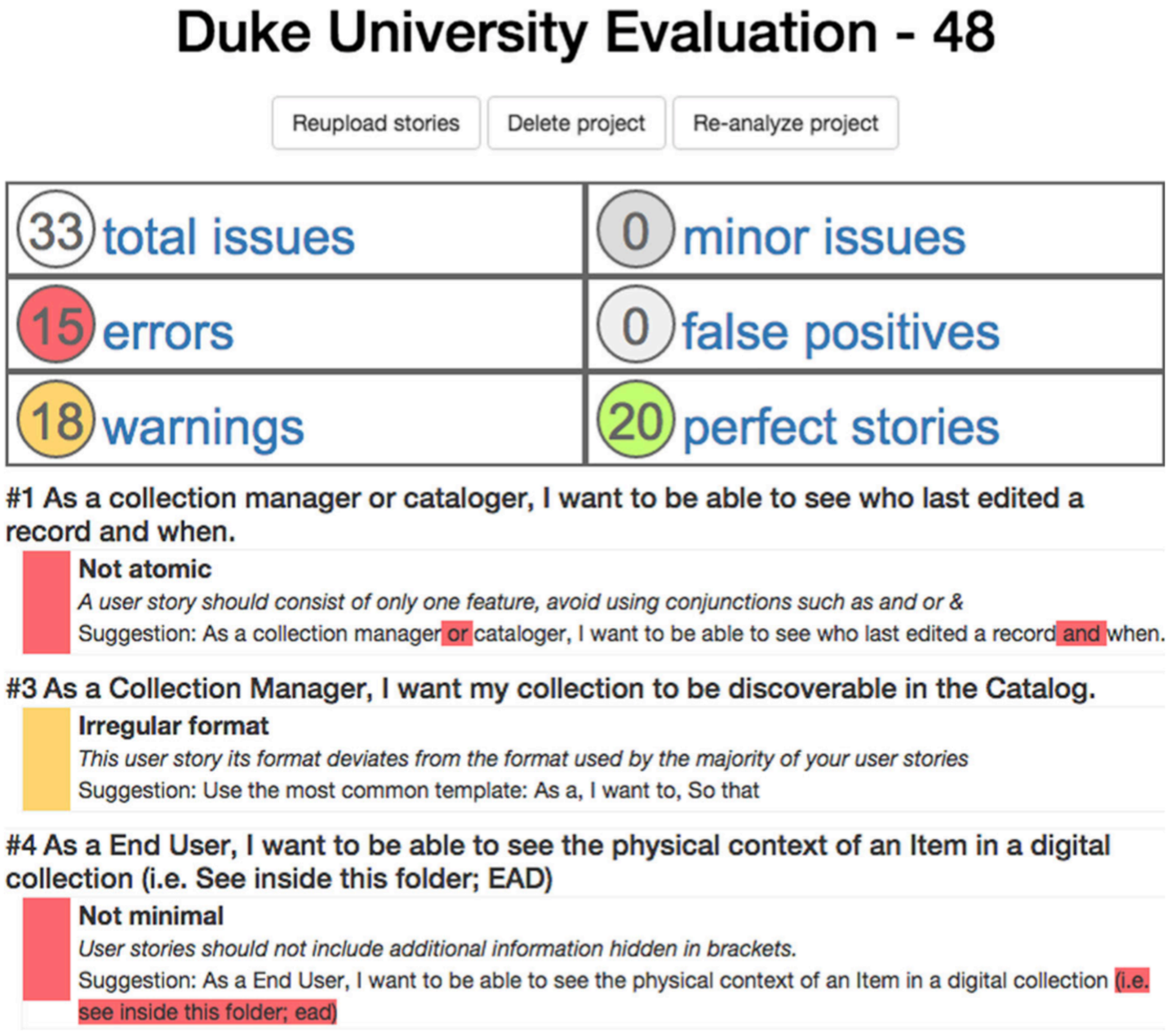

AQUSA
Synonyms
Homonyms
Corrections
Ontologies
Enhancer
User stories
Linguistic parser
User story base
Report generator
Error report
Analyzer
Atomic
Minimal
Independent
Uniform
Unique

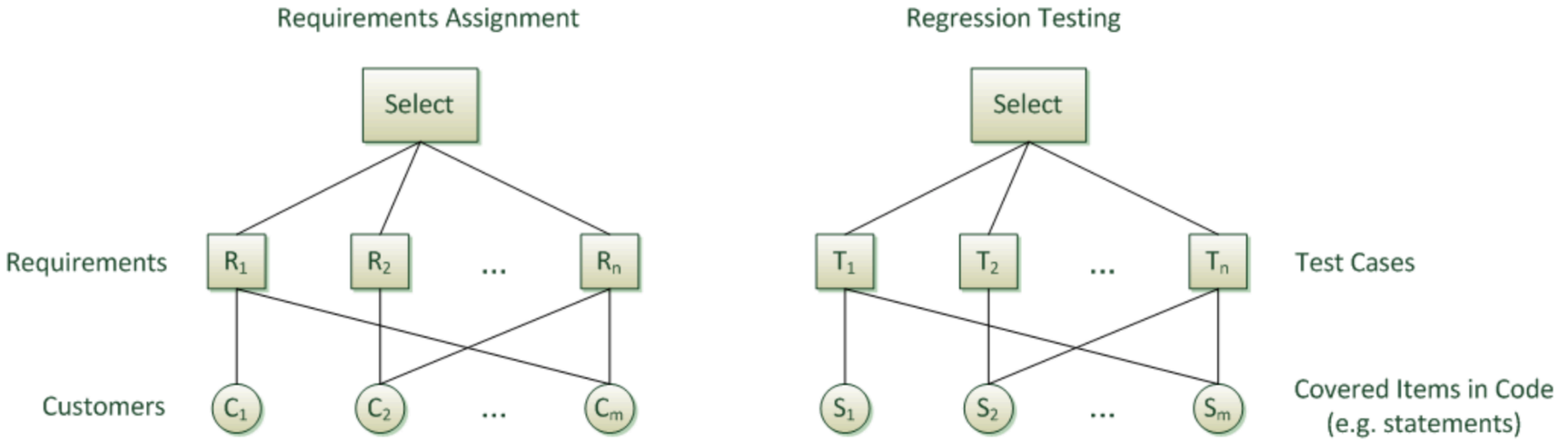
Fig. 5 Example report of a defect and warning for a story in AQUSA



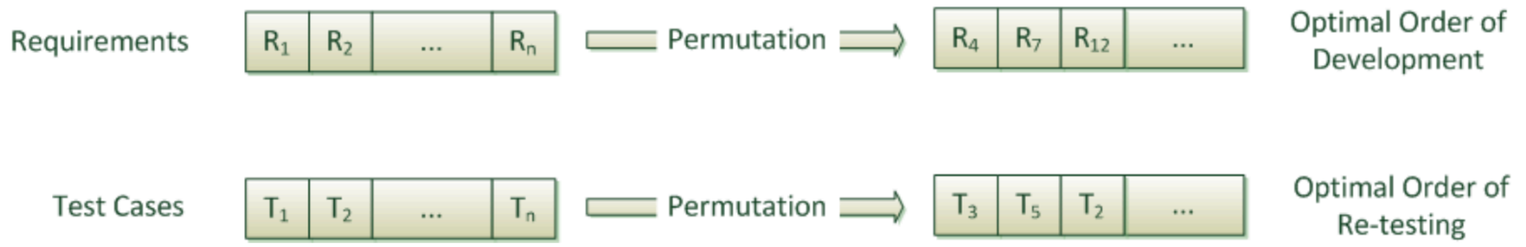
ML for the Next Release Problem

- Prioritizing requirements over many dimensions (cost, importance, dependencies, safety, etc.) is a multi-criteria optimization problem
  - Thus, no good optimal solutions exist (assuming P!=NP)
- Heuristic search can be one approach that is effective for the Next Release Problem (Bagnall, 2001).
- Each customer, *i*, will have a set of requirements, Ri \subset R; and a weight, wi in positive integers; which is a measure of the customer's importance to the company.

Selection Problems



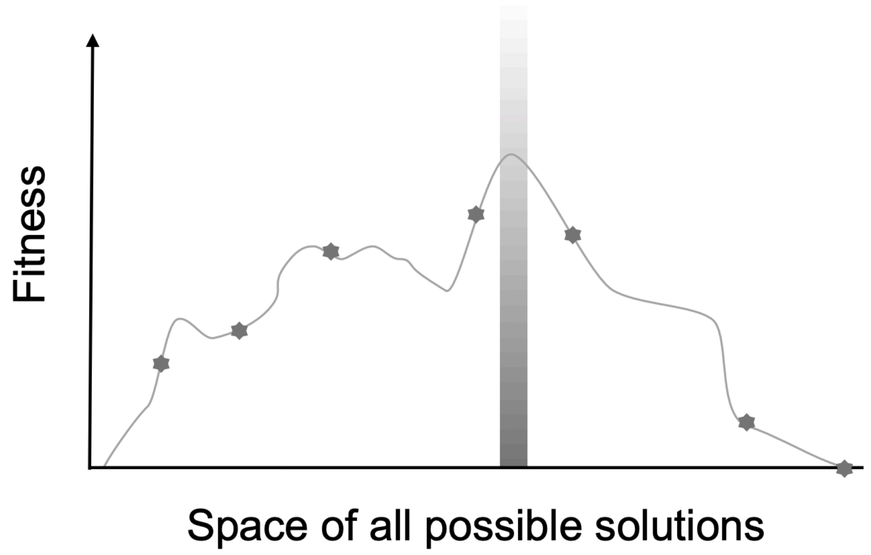
Prioritization Problems



From: "Search Based Software Engineering: Techniques, Taxonomy, Tutorial" by Mark Harman[1], Phil McMinn[2], Jerffeson Teixeira de Souza[3], and Shin Yoo[1]

Search Based Optimization

- Define a fitness function (how you know you are getting closer to the solution)
  - Often common metrics in SE, such as test coverage
- Define the problem (e.g., a graph)
- Choose the algorithm:
  - Random, Hill climbers, Simulated Annealing e
  - Genetic algorithm



Randomly generate or seed initial population P
Repeat
    Evaluate fitness of each individual in P
    Select parents from P according to selection mechanism
    Recombine parents to form new offspring
    Construct new population P' from parents and offspring
    Mutate P'
    $P \leftarrow P'$
Until Stopping Condition Reached

Fig. 8. High level description of a Genetic Algorithm, adapted from McMinn [65]

E.g. certain number of rounds, or desired fitness

Crossover operation and Pareto dominance