A Refined Model of Ill-definedness in Project-Based Learning

EduSymp @ MoDELS 2022, 25 October 2022

Arthur Rump, Vadim Zaytsev

UNIVERSITY OF TWENTE.

What’s in the Paper?

- https://grammarware.net/text/2022/illdefined.pdf
- https://doi.org/10.1145/3550356.3556505
- data collection + analysis
- here let’s focus on context!

A Refined Model of Ill-definedness in Project-Based Learning

Arthur Rump
Computer Science
University of Twente
The Netherlands
a.h.j.rump@student.utwente.nl

Vadim Zaytsev
Formal Methods and Tools
University of Twente
The Netherlands
vadim@grammarware.net

ABSTRACT

Project-based courses are crucial to gain practically relevant knowledge in modelling and programming education. However, they fall into the “ill-defined” domain: there are many possible solutions; the quality of a deliverable is subjective and not formally assessable; reaching the goals means designing new artefacts and analysing new information; and the problem cannot always be divided into independent tasks. In this paper, we refine the existing two-dimensional (verifiability and solution space) classification of ill-defined classes of problems, contemplate methods and approaches for assessment of projects, and apply the model to analyse two study units of two different computer science programmes.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**.

KEYWORDS

learning objectives

ACM Reference Format:
Arthur Rump and Vadim Zaytsev. 2022. A Refined Model of Ill-definedness in Project-Based Learning. In *ACM/IEEE 25th International Conference on Model Driven Engineering Languages and Systems (MODELS ’22 Companion), October 23–28, 2022, Montreal, QC, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3550356.3556505

1 INTRODUCTION

Project-based learning is a student-centred form of learning based on the constructivist ideas that learning is context-specific, that students learn best when they are actively involved in the learning process and that learning happens through social interaction and the sharing of knowledge [9]. Although not every implementation of a project means that project-based learning is applied, it is also difficult to use projects purely for summative assessment because students will almost always learn new things when working on a complex project. This means that some of the benefits of project-based learning are likely to occur to some degree in any course that uses a project.

A project in project-based learning has two essential characteristics: there is a *driving question*, often in the form of a problem to be solved, and the learning activities result in a set of artefacts or products, which represent students’ solutions and reflect their knowledge [4, 10]. The question and activities can be determined by students or teachers, but it is important that the question is not so constrained that the outcomes are predetermined. Blumenfeld et al [4, p.372] note that “students’ freedom to generate artefacts is critical, because it is through this process of generation that students construct their knowledge.”

This freedom in generating artefacts that become the solution, eventually delivered to the teachers for grading, leads to many issues in assessment of that solution. Such problems are called “ill-defined”, and this ill-definedness of a project is crucial to let students construct their knowledge and thus learn. We will recall the definition of ill-defineness and link it to our situations in § 2. Then, in § 3, we will introduce two study units of two different programs at our university, each being the first opportunity for corresponding students to model an entire software system by themselves by applying principles of object-oriented design. In § 4 we crystallise lessons learnt from analysing these two into concrete refinements on the existing model of ill-definedness. In § 5, we apply the resulting framework to the two study units we just introduced. The paper ends with § 7 which draws some conclusions.

2 ILL-DEFINEDNESS

A word that is commonly used for problems that do not have a definite solution is **ill-defined**. Such problems have an *indefinite endpoint*, meaning that determining if the goal has been reached is complex and imprecise, and it is one of three criteria Simon [15] (as cited in [5], we lack access to the original) describes for calling a problem ill-defined. The other two features they find are an *indefinite starting point*, meaning that the problem description is vague or incomplete, and *unclear strategies* for finding a solution. Wherever a project description mentions “good” solutions or talks about a free selection of additional features, we see ill-definedness. Additionally, software engineering has, in general, no single strategy for finding a solution: it always involves some creativity that starts to manifest at use case diagrams and persists till the last line of code.

Lynch et al describe five features of an ill-defined domain [13]:

- *There are multiple solutions, and which one is better is partly subjective.* This is certainly the case for project-based learning: every group will likely have very different solutions, and which follows a better style is to some degree a matter of taste.
- *There is no formal theory for determining a problem’s outcome and testing its validity.* There is no formal theory of modelling and programming that can derive a correct and valid program

MODELS ’22 Companion, October 23–28, 2022, Montreal, QC, Canada
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
This is the author’s version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ACM/IEEE 25th International Conference on Model Driven Engineering Languages and Systems (MODELS ’22 Companion), October 23–28, 2022, Montreal, QC, Canada*, https://doi.org/10.1145/3550356.3556505.

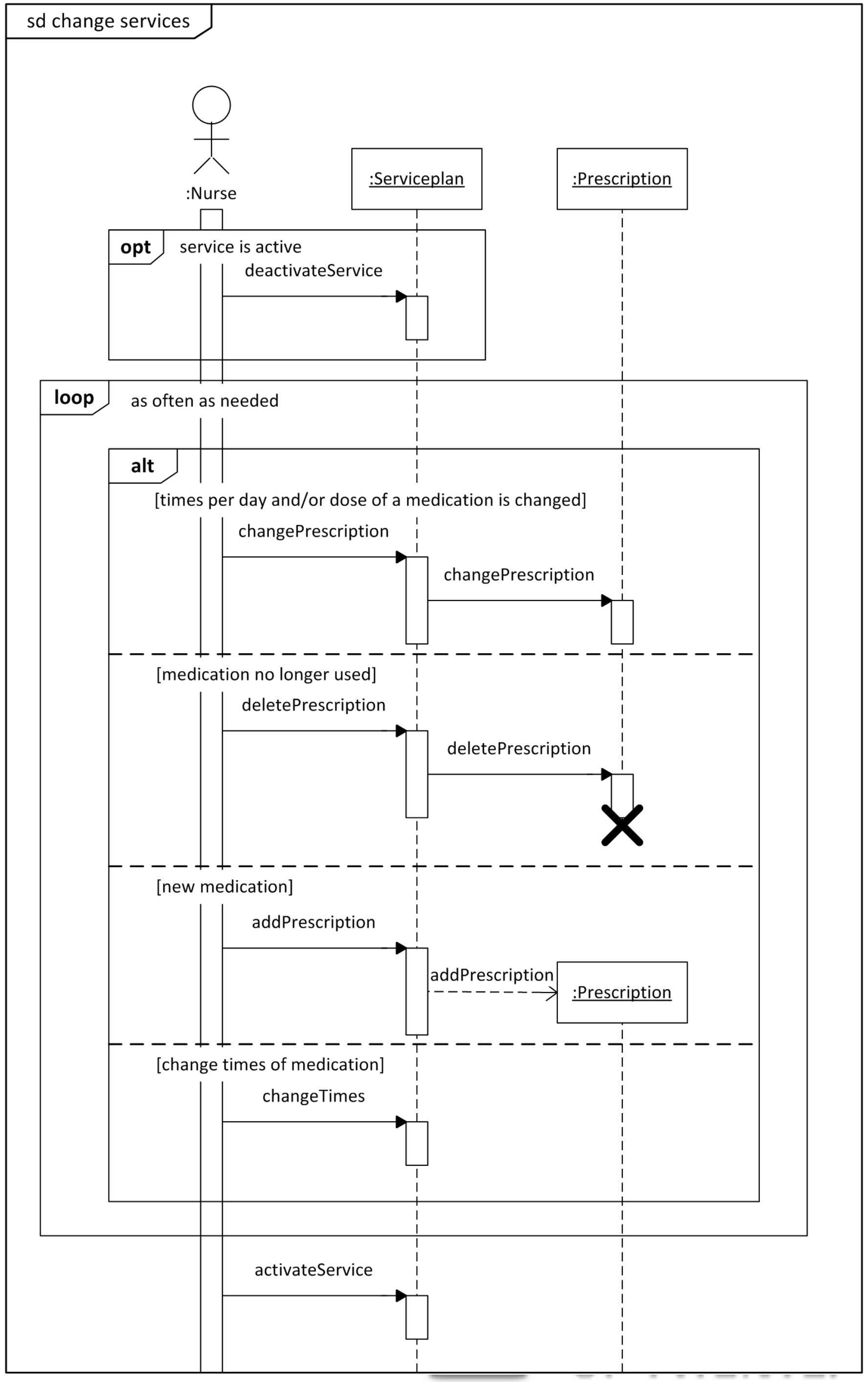
UNIVERSITY OF TWENTE.

Well-defined

- Understood problem
- Concrete verifiable concepts
- Subdivide and solve
- Some solutions are incorrect
- Empirical validation
- Known mathematical foundation
- Engineered artefacts

Ill-defined

- Incomplete information
- Abstract concepts
- Dependent subproblems
- Multiple solutions
- Subjective criteria
- Informal validation
- Artefact design



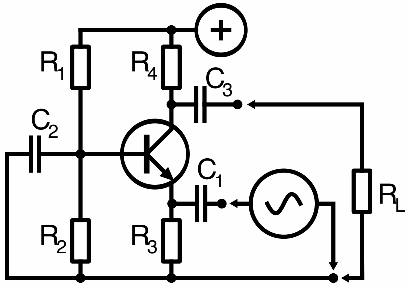
Ill-defined == Bad

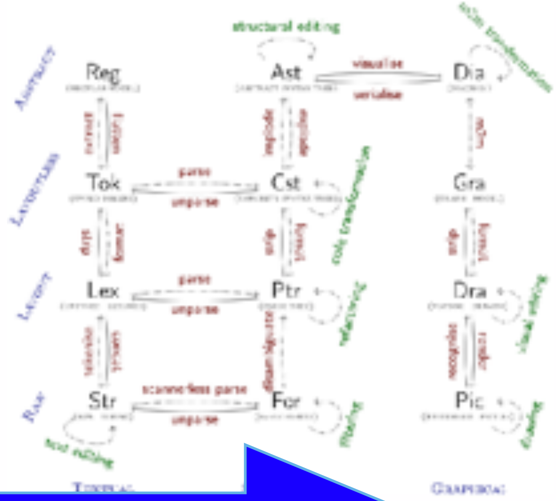
- Incomplete information ⇒ ambiguous/incomplete
- Abstract concepts ⇒ no absolute/teachable definitions
- Dependent subproblems ⇒ modularity impossible/hard
- Multiple solutions ⇒ predictability limited
- Subjective criteria ⇒ assessment unreliable
- Informal validation ⇒ no formal theory to support
- Artefact design ⇒ tasks undefined

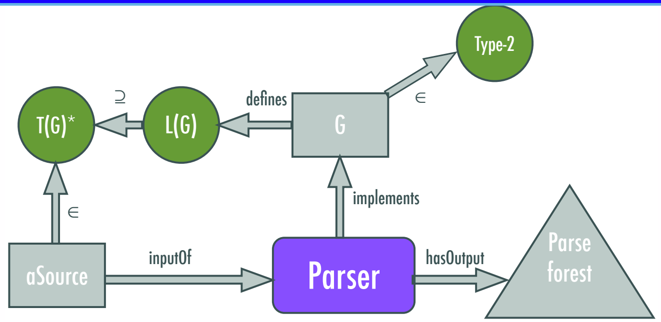
Ill-defined $==$ Good

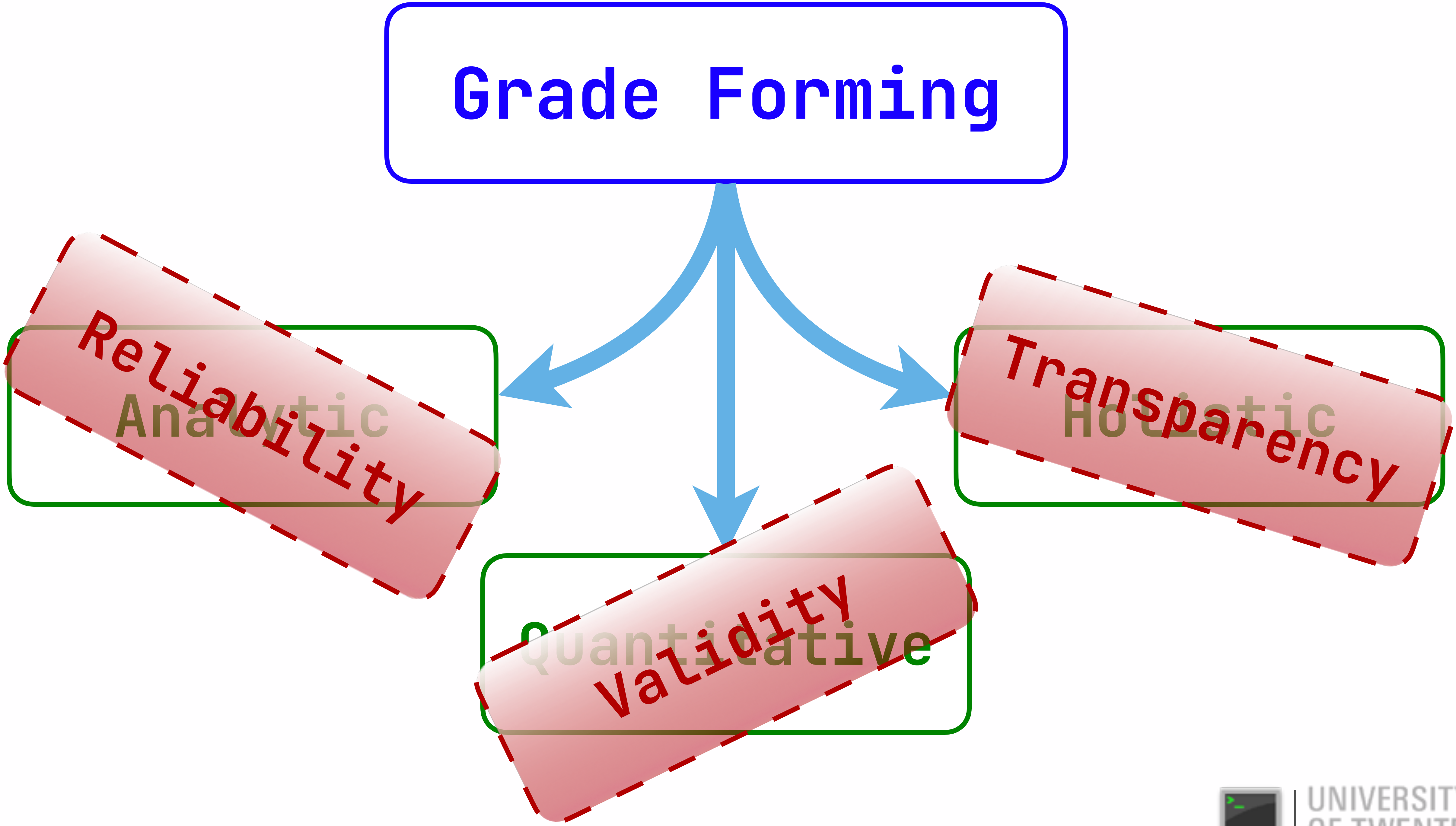
- Incomplete information ⇒ possible gamification
- Abstract concepts ⇒ contrasting cases
- Dependent subproblems ⇒ modularisation is a skill
- Multiple solutions ⇒ entertaining for the teacher
- Subjective criteria ⇒ descriptive feedback/reflection
- Informal validation ⇒ space for creativity
- Artefact design ⇒ industry-relevant skill

Definedness is a Spectrum









UNIVERSITY OF TWENTE.

Great variability of solution strategies		
Known number of strategies		
One good strategy, several implementations		
One solution		
	Verifiable	Unverifiable

model from https://doi.org/10.1109/tlt.2013.16

	Objective	Subjective, with rubric	Subjective, no rubric	Free choice
Great variability of solution strategies				
Known number of strategies				
One good strategy, several implementations				
One solution				

model from https://doi.org/10.1145/3550356.3556505

	Objective	Subjective, with rubric	Subjective, no rubric	Free choice
Great variability of solution strategies				
Known number of strategies				
One good strategy, several implementations				
One solution				

empty?

model from https://doi.org/10.1145/3550356.3556505

	Objective	Subjective, with rubric	Subjective, no rubric	Free choice
Great variability of solution strategies				
Known number of strategies				
One good strategy, several implementations				
One solution				

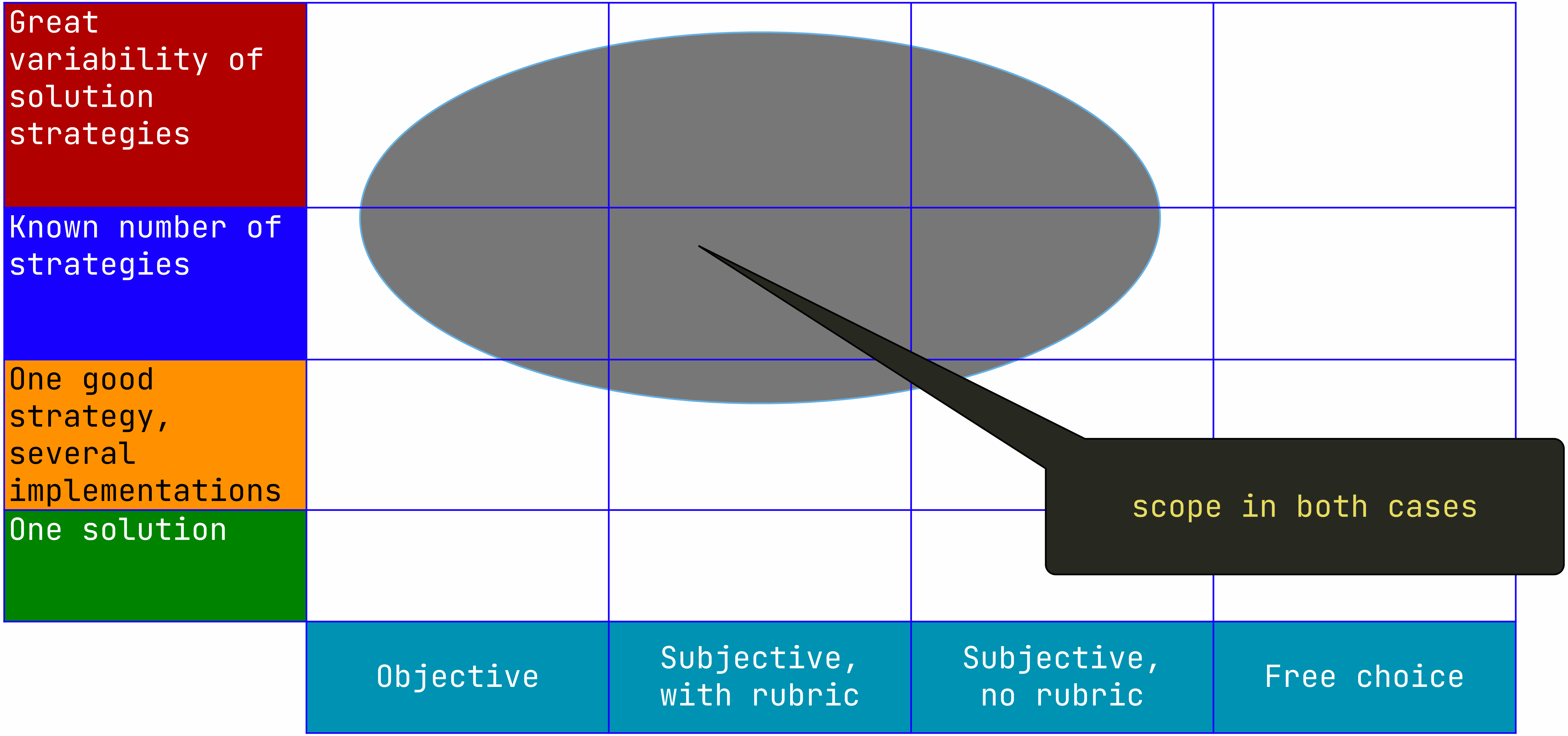
empty in Algorithms for Creative Technology

model from https://doi.org/10.1145/3550356.3556505

Great variability of solution strategies				
Known number of strategies				
One good strategy, several implementations				
One solution				
	Objective	Subjective, with rubric	Subjective, no rubric	Free choice

correctness in Programming of Software Systems

model from https://doi.org/10.1145/3550356.3556505



model from https://doi.org/10.1145/3550356.3556505

Great variability of solution strategies				
Known number of strategies				
One good strategy, several implementations				
One solution				
	Objective	Subjective, with rubric	Subjective, no rubric	Free choice

quality in both cases

model from https://doi.org/10.1145/3550356.3556505

- project-based learning
- embracing ill-definedness
- model is refined on one dimension
  - before: verifiable, unverifiable
  - after: objective, rubric, subjective, free
- more in the paper!

https://www.arthurrump.com/ & https://grammarware.net/