Quantum Software Development: A Survey

Results

Introduction

- Over the last few years, quantum computing has been growing at an exponential pace. Every day, new techniques, frameworks, modeling, and programming languages are emerging that aim to facilitate the development of quantum software (which is key to achieve the promising applications of quantum computing). However, which of these are actively used and the degree of satisfaction of researchers and developers regarding these quantum software frameworks and languages is not known. To address this, we conducted a survey to characterize which modeling tools and which quantum programming languages are used during the quantum software lifecycle. Both researchers in academia and developers in the industry were surveyed and a total of 57 responses were collected.

Authors

Luis Jiménez-Navajas

Fabian Bühler

Frank Leymann

Ricardo Pérez-Castillo

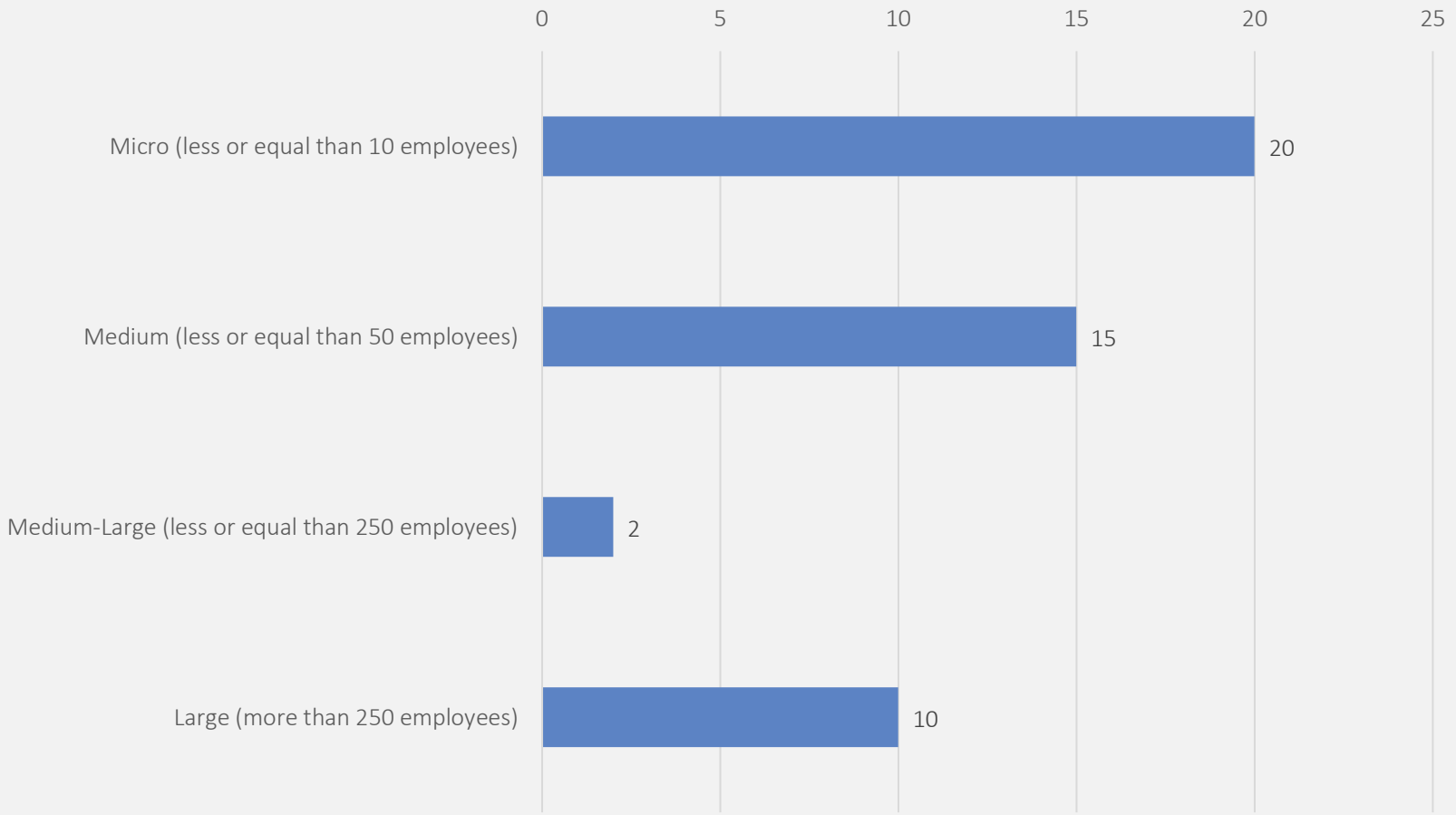
Mario Piattini

Daniel Vietz

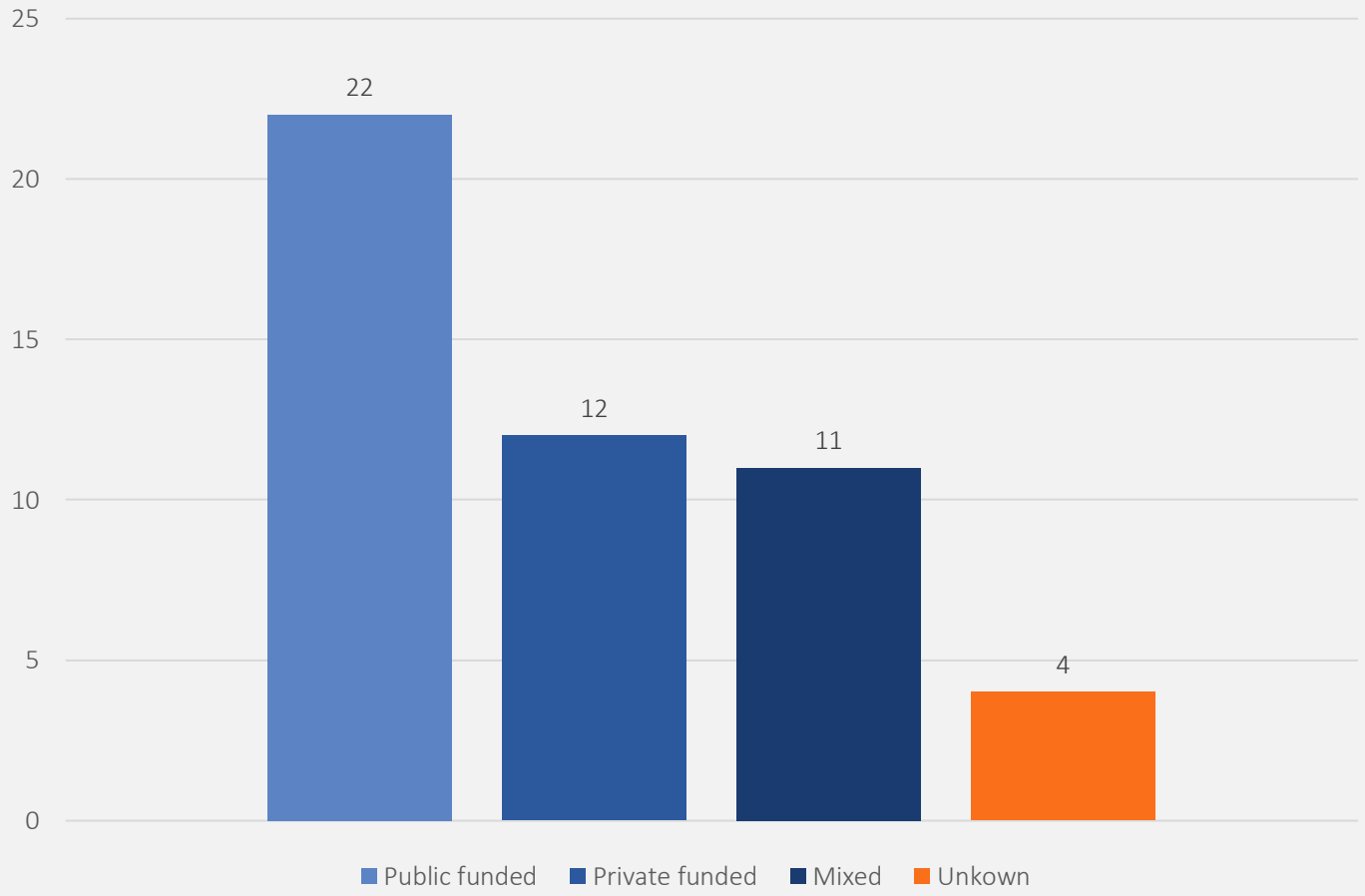
Results obtained from the survey

Respondent’s profile

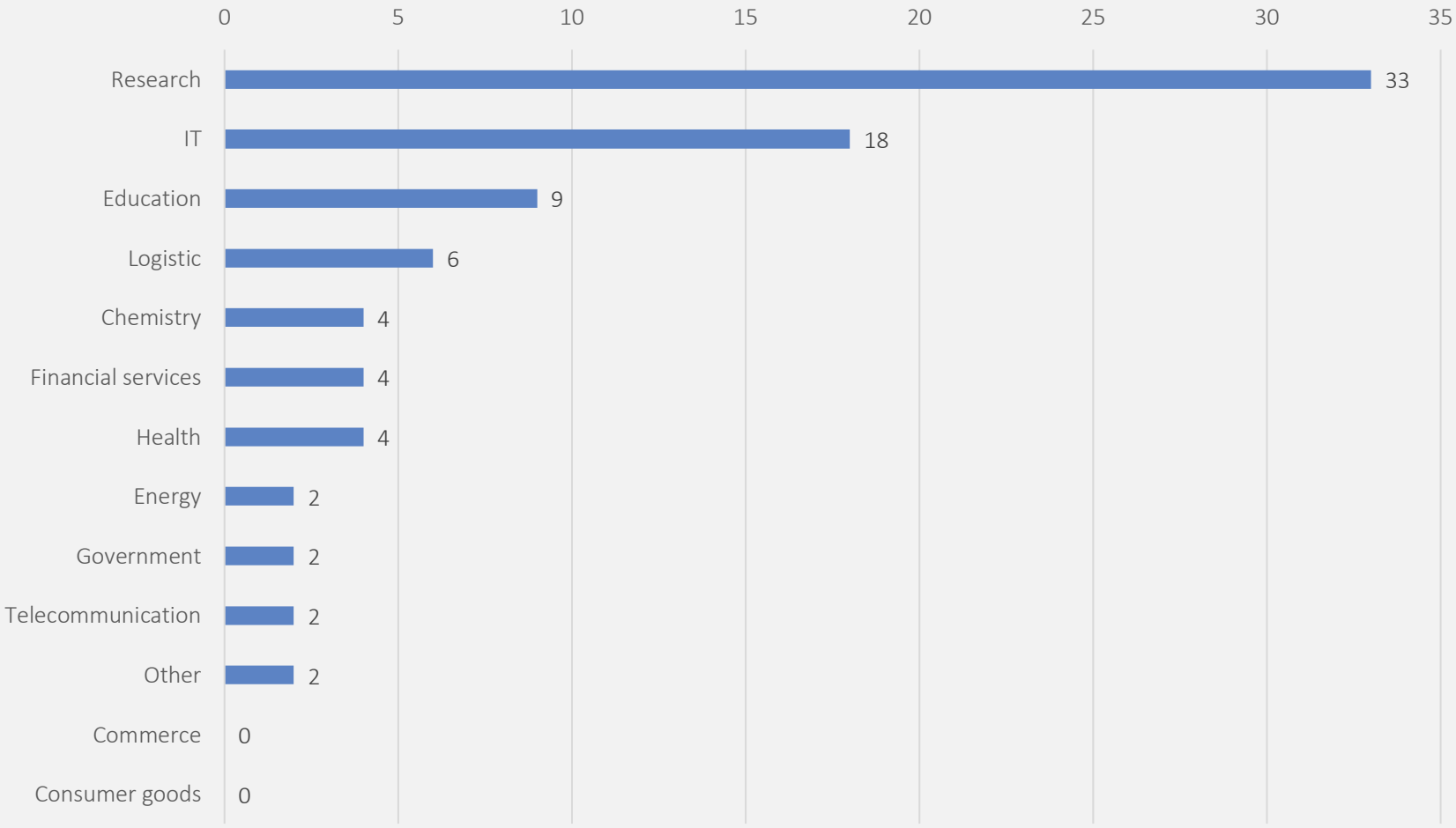
Size of the organization



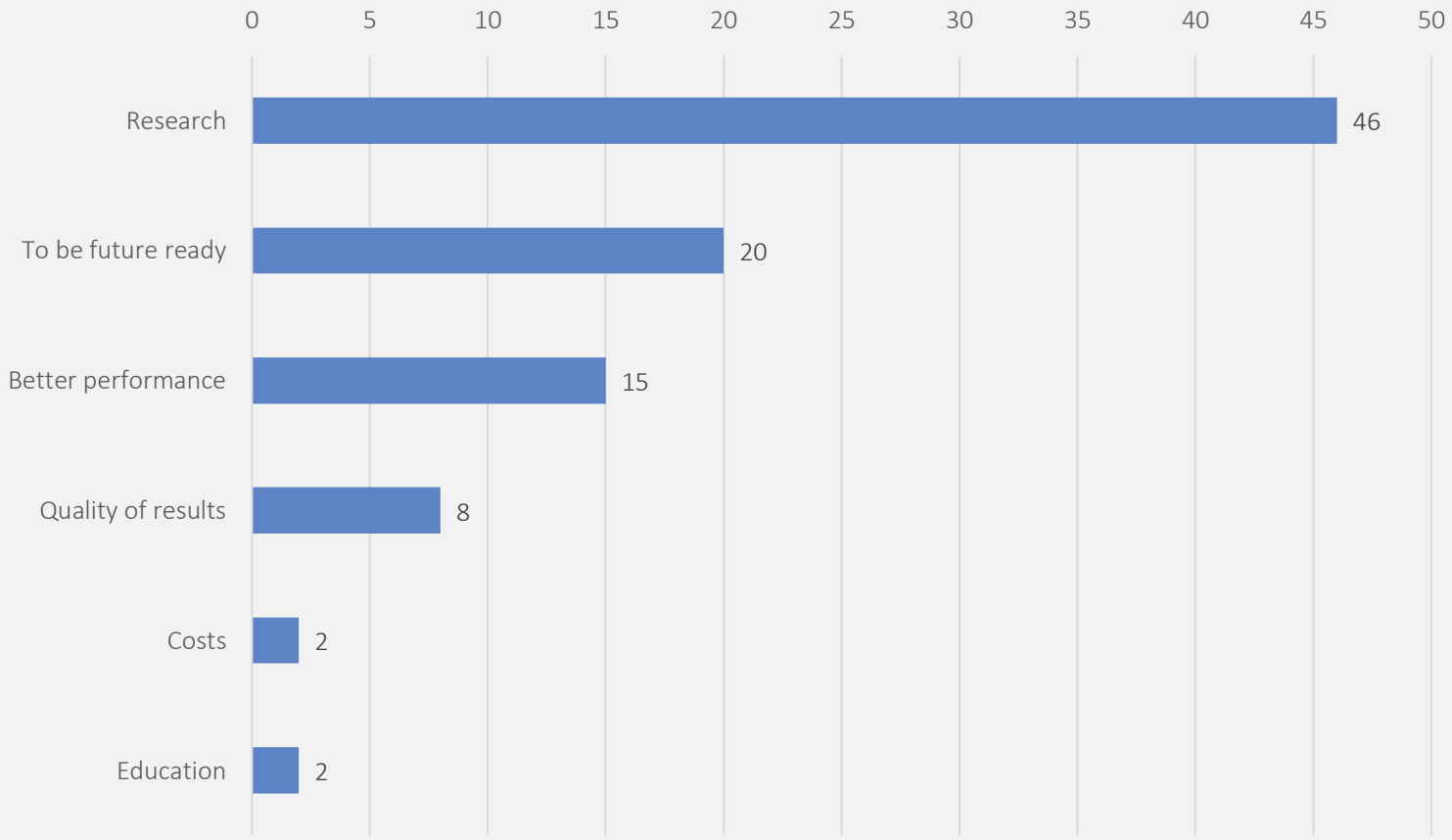
Organization's type



Target sector of the hybrid software



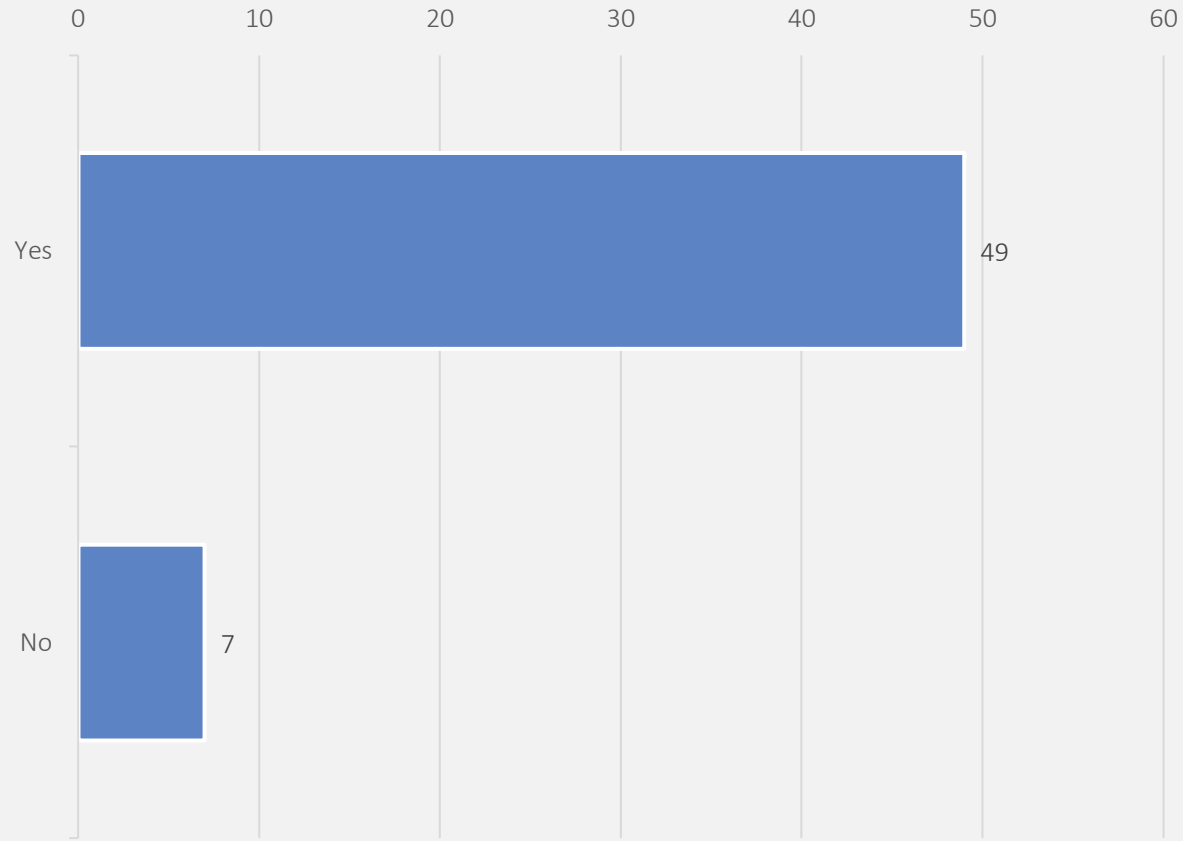
Organization's motivation regarding the development of quantum software



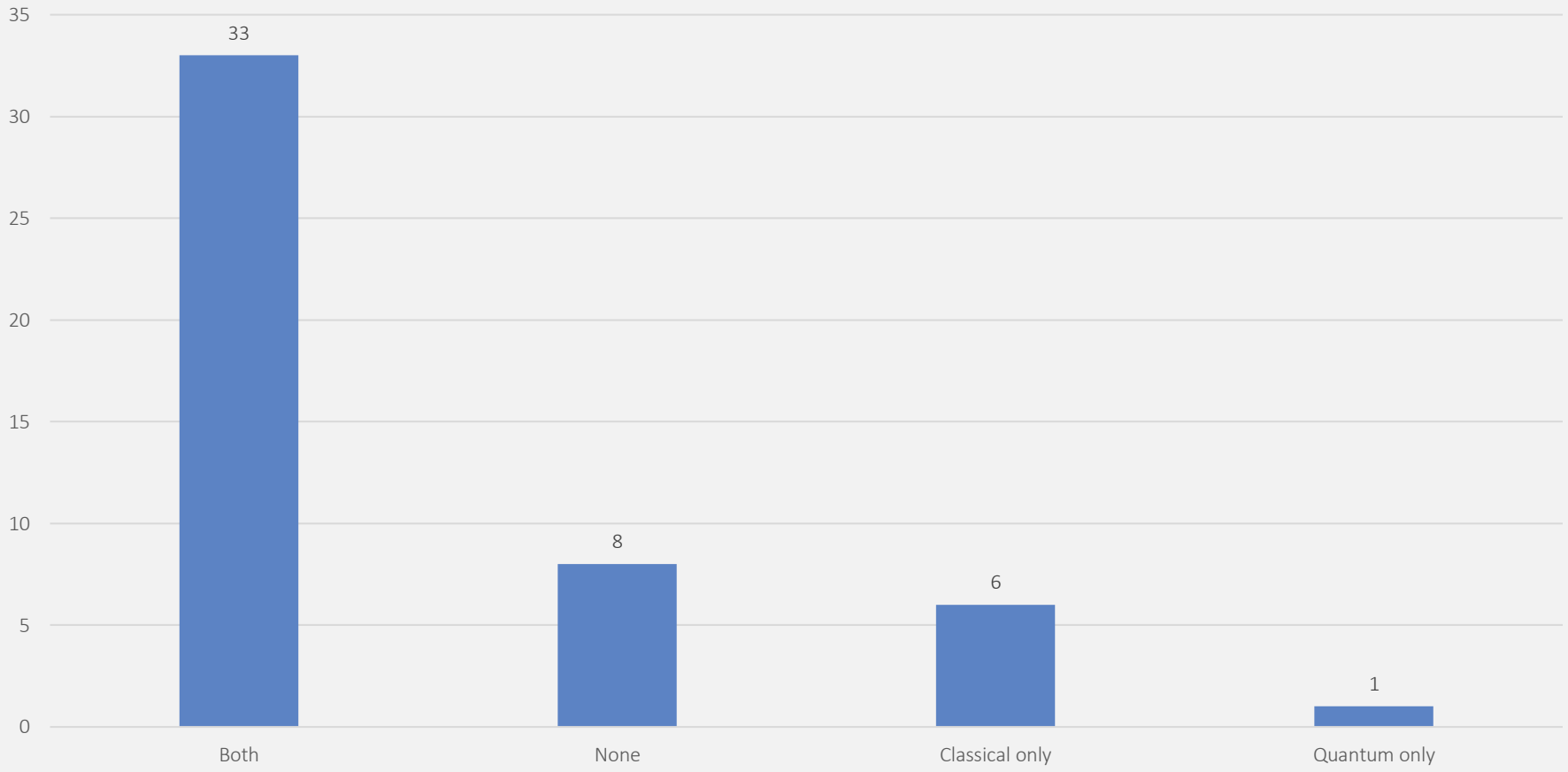
Results obtained from the survey

Goal 1: Quantum software modeling

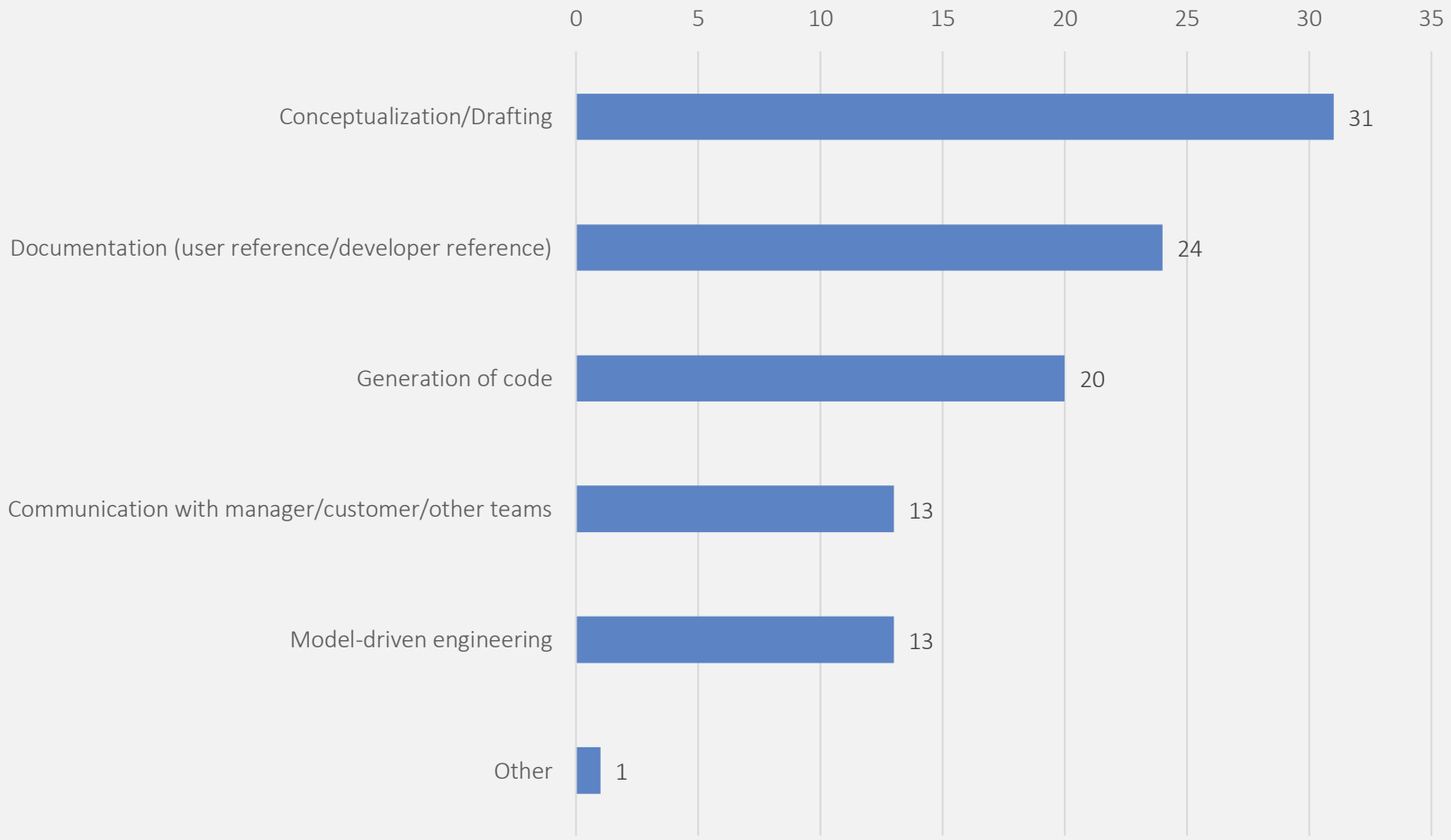
Have you developed quantum software in the past?



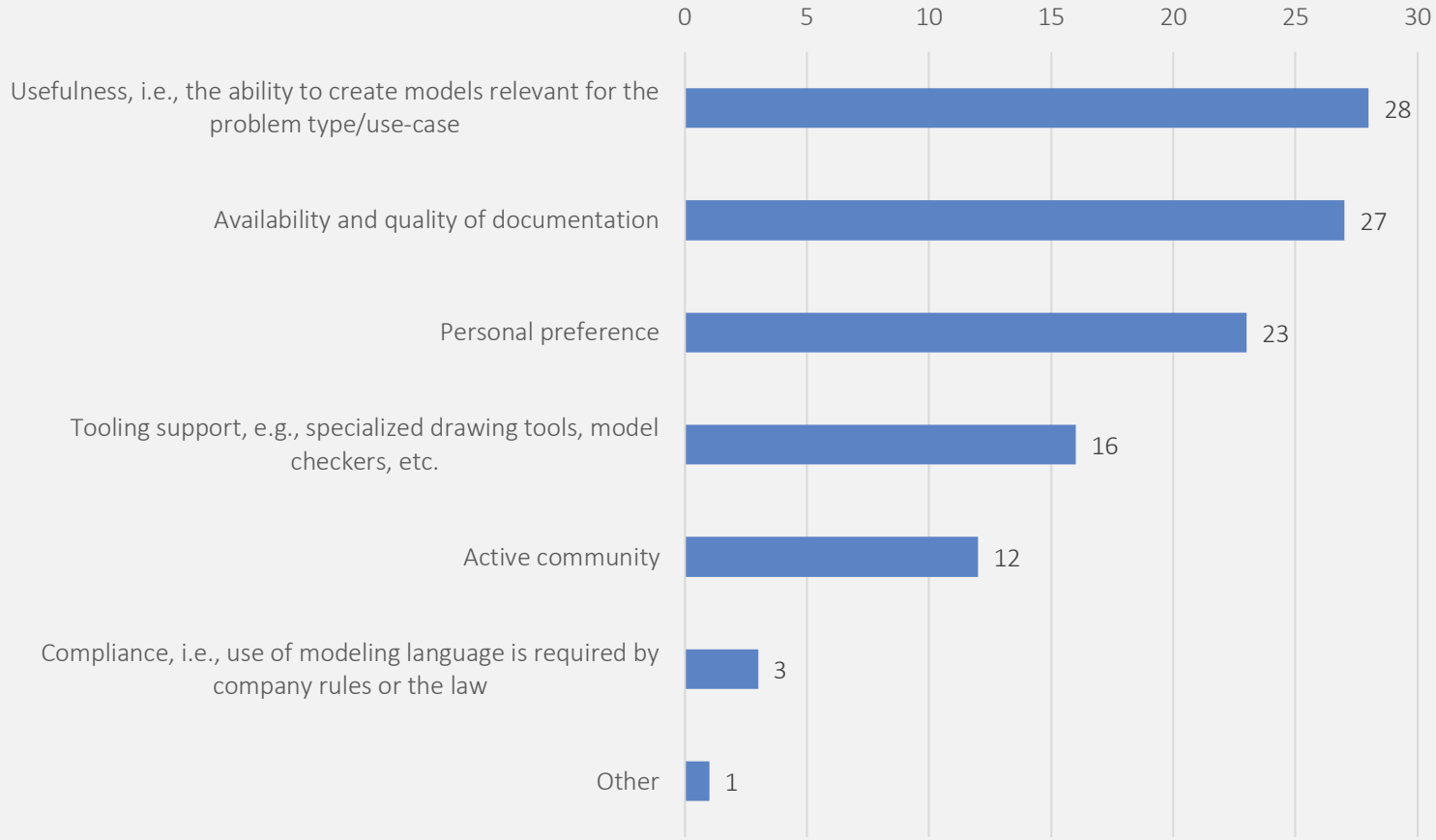
Software components modeled



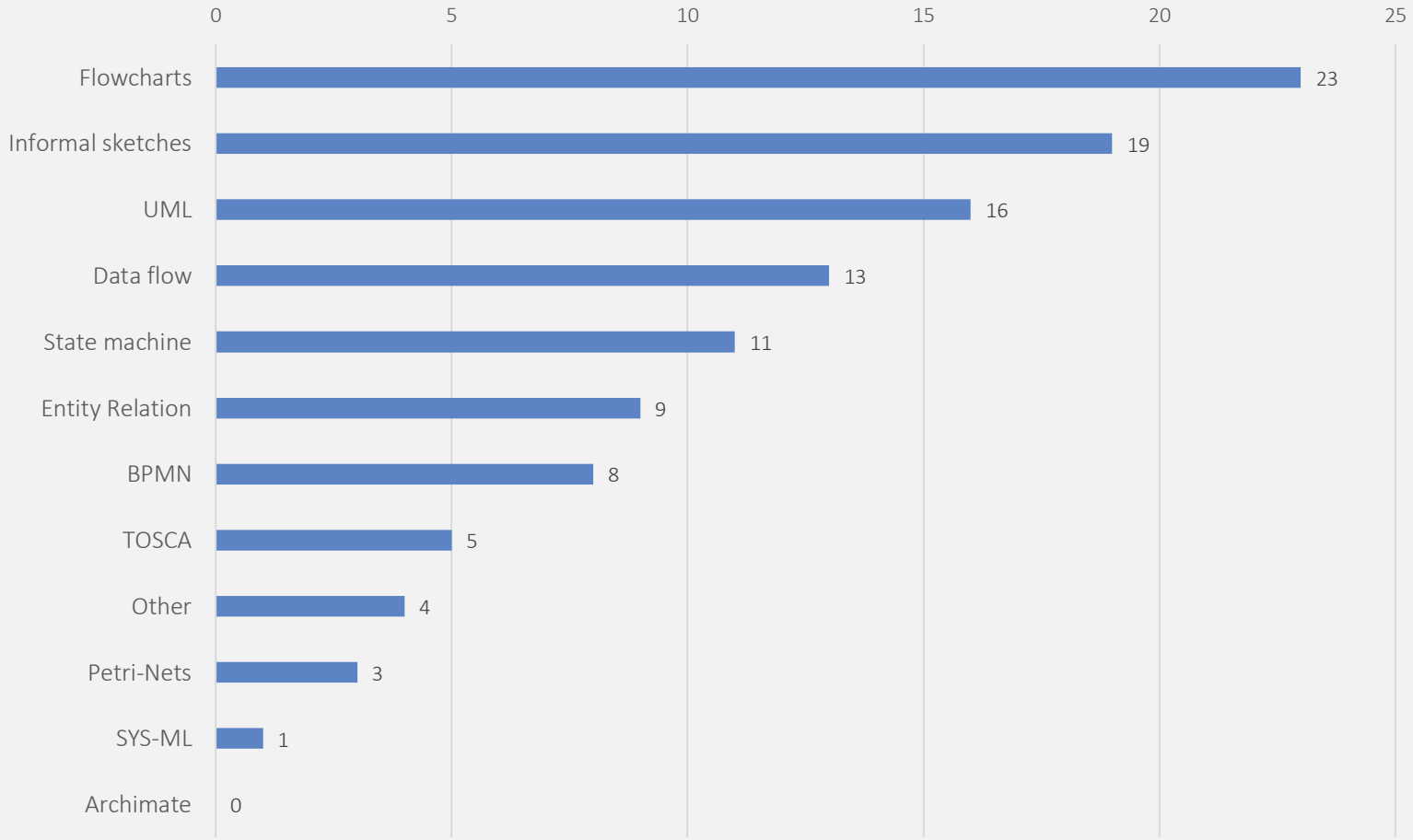
Use of models



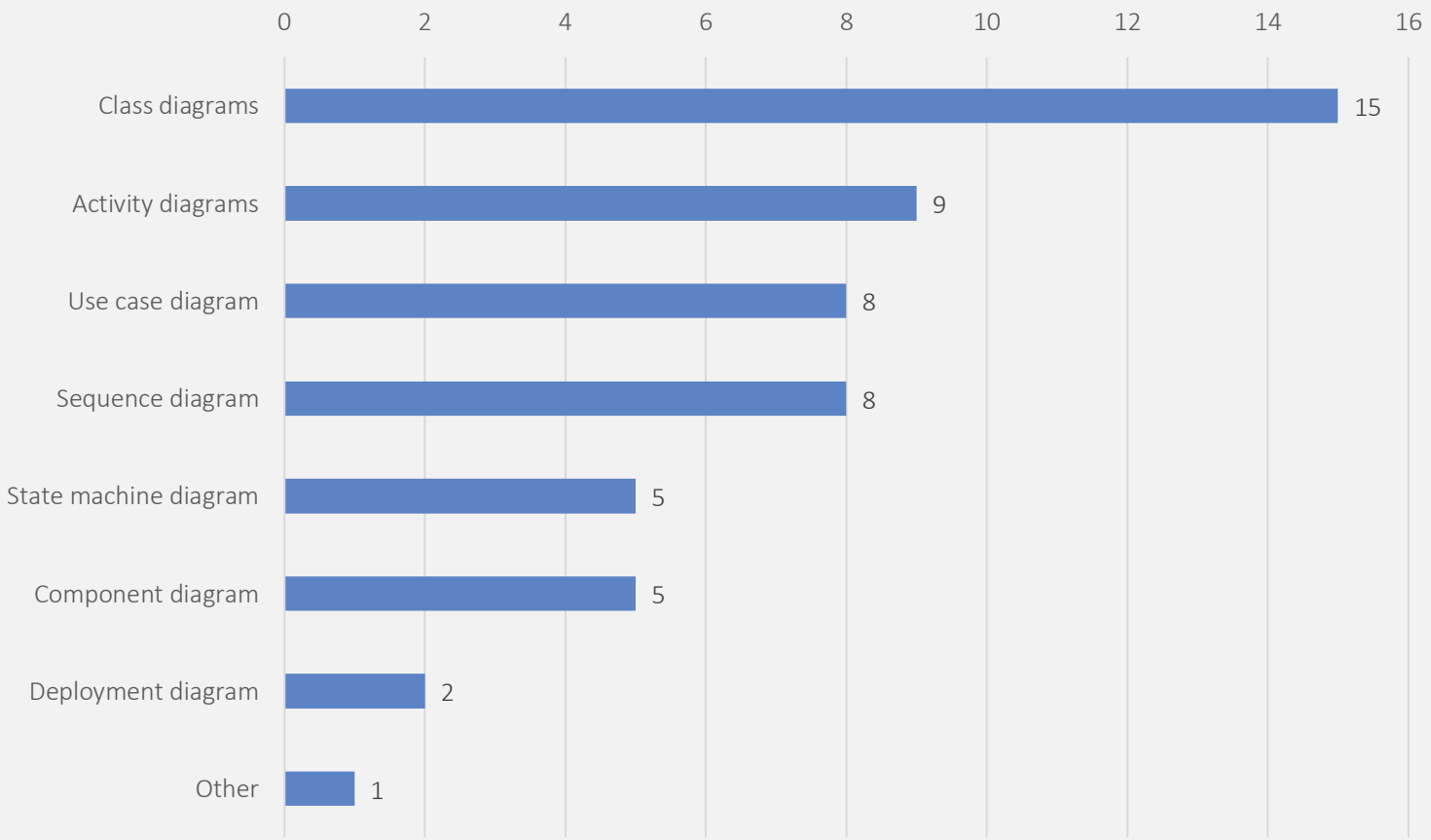
Criteria to decide between modeling languages or tools for modeling



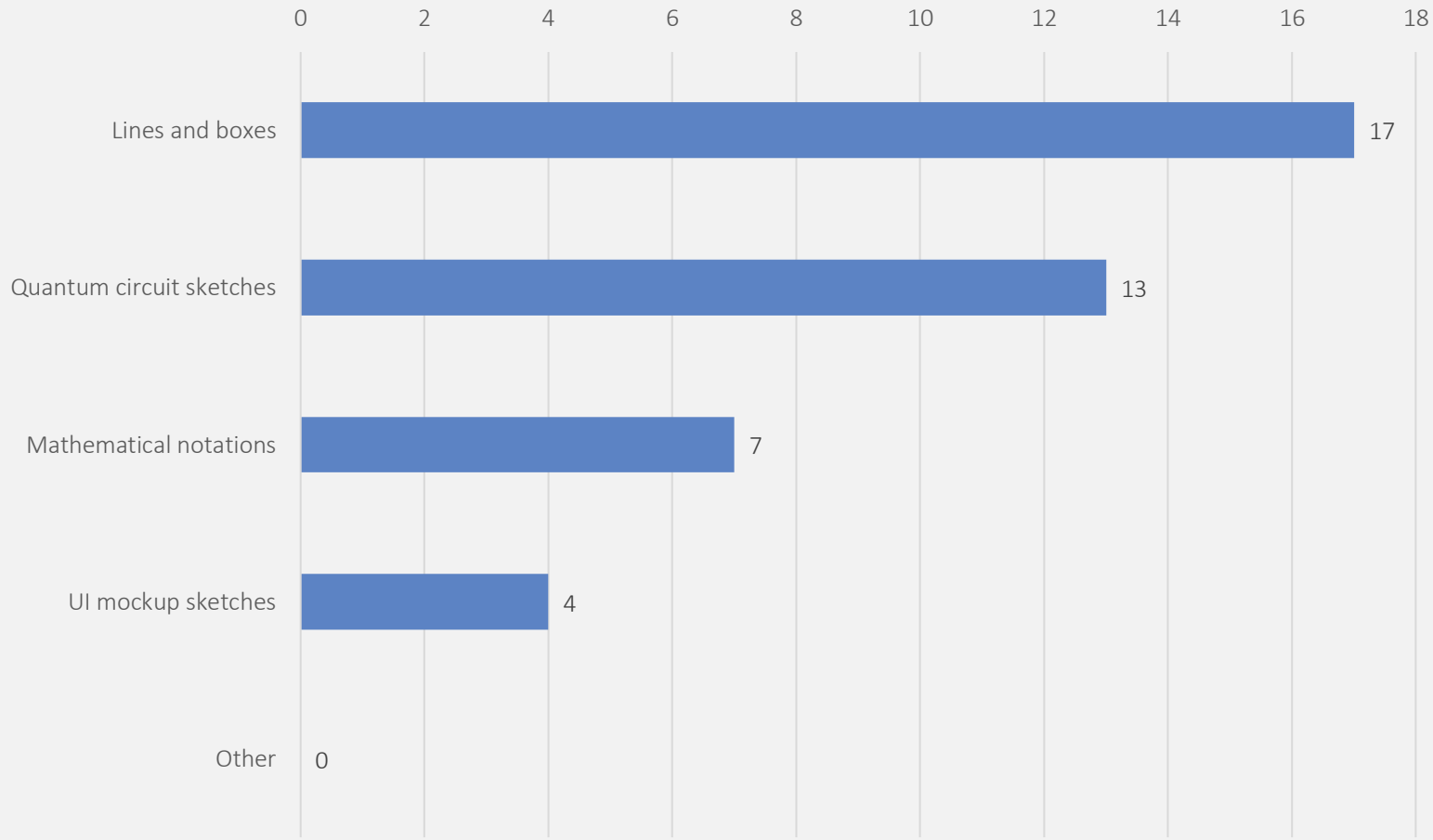
Modeling languages or diagram types used during the development of quantum software



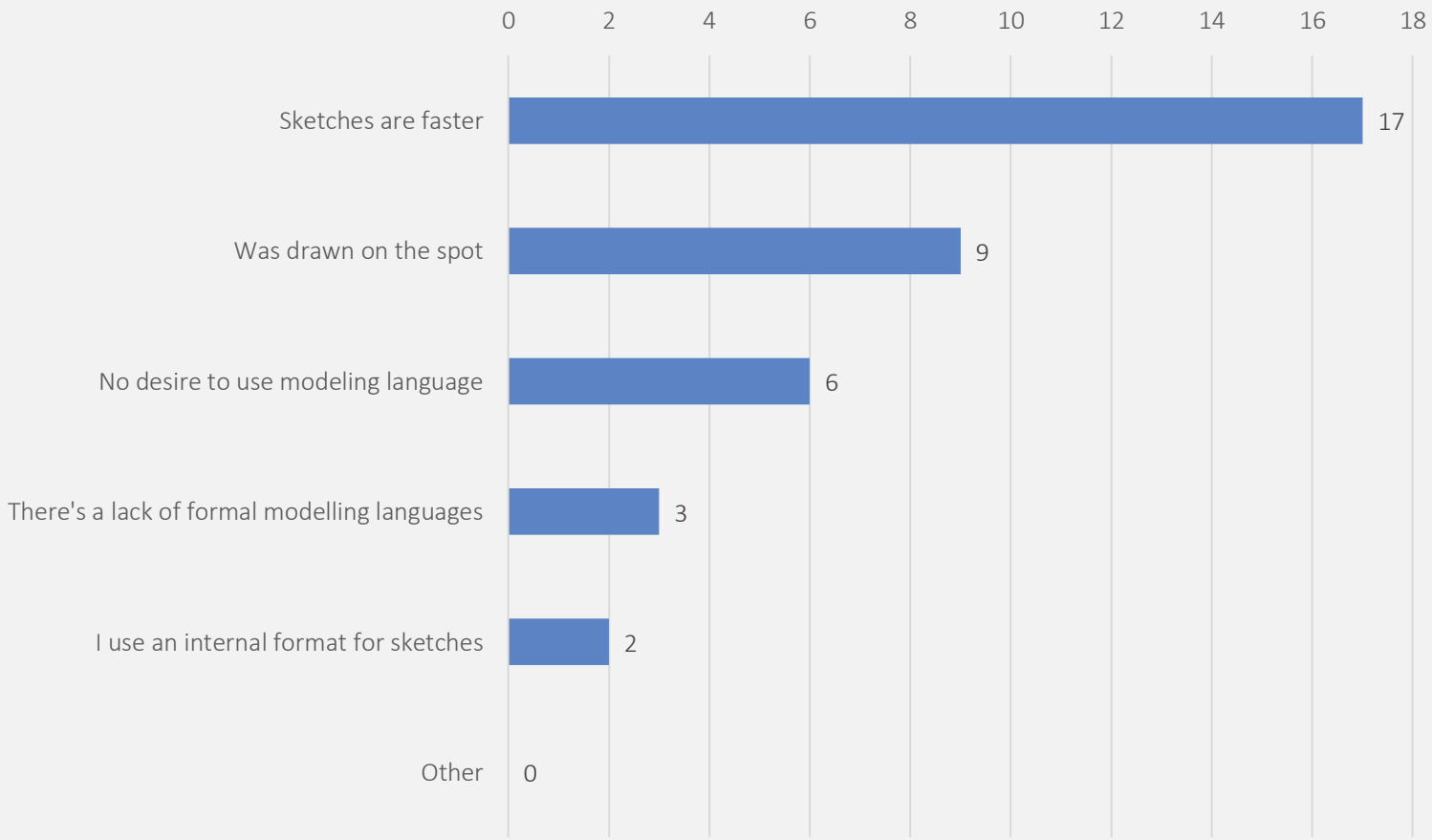
Type of UML diagrams used



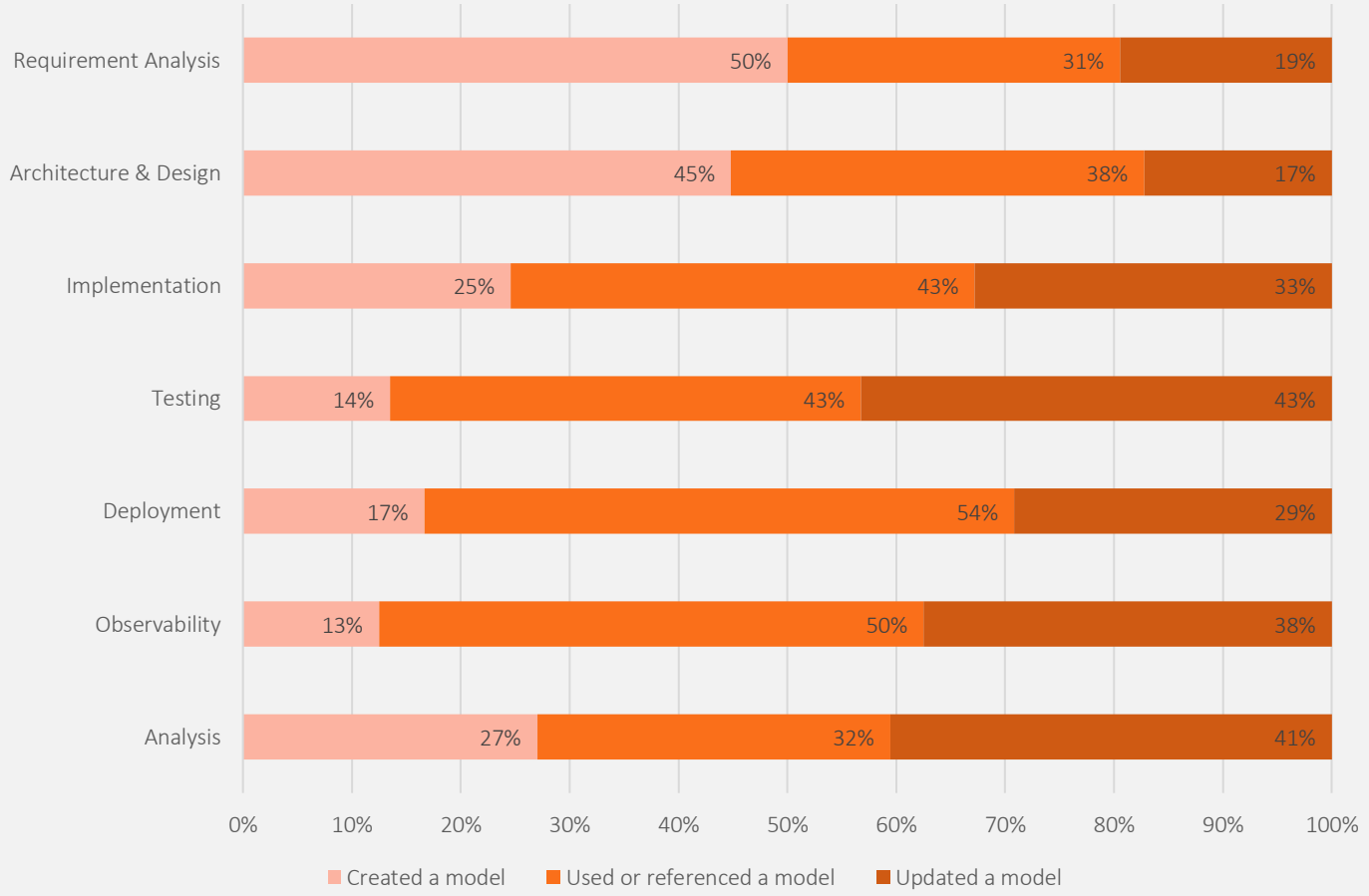
Type of informal sketches used



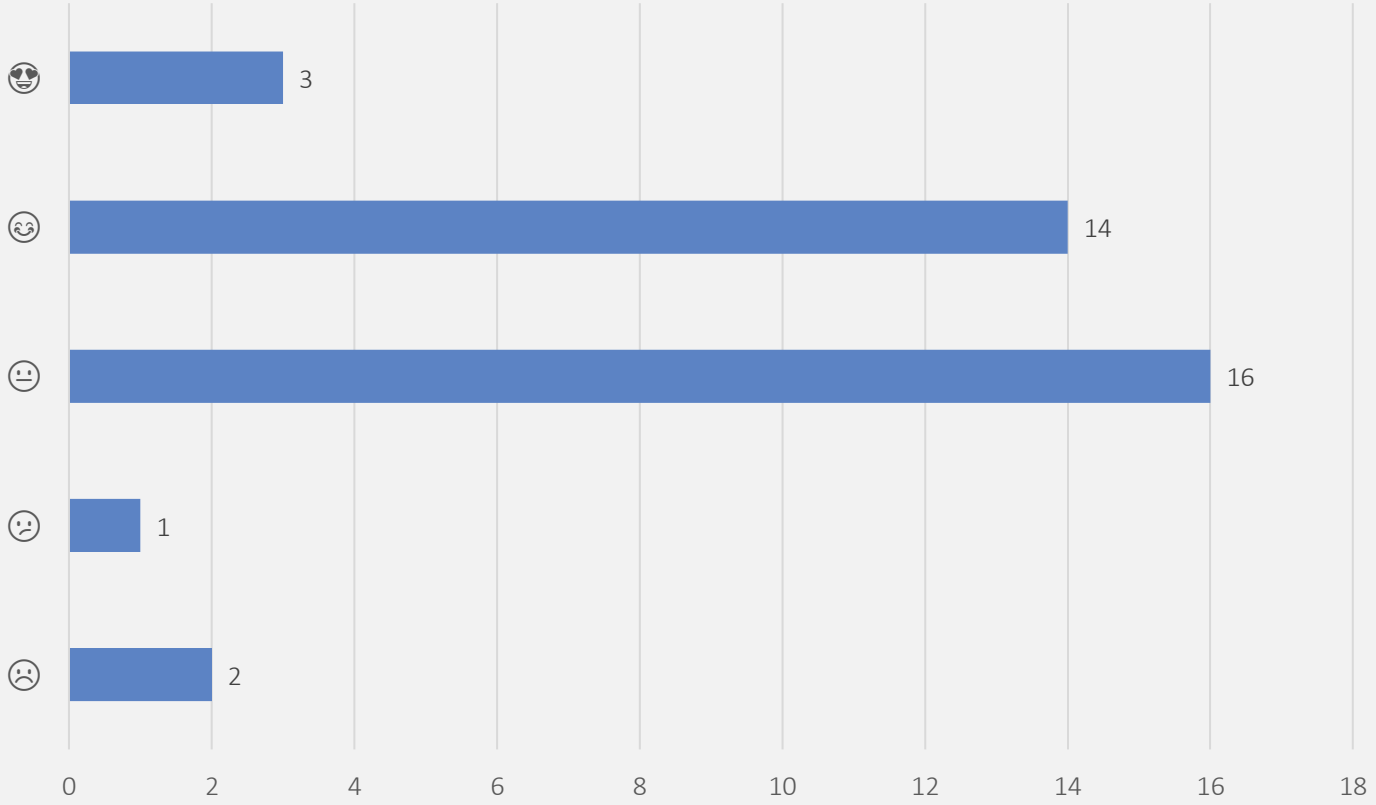
Reasons to employ informal sketches



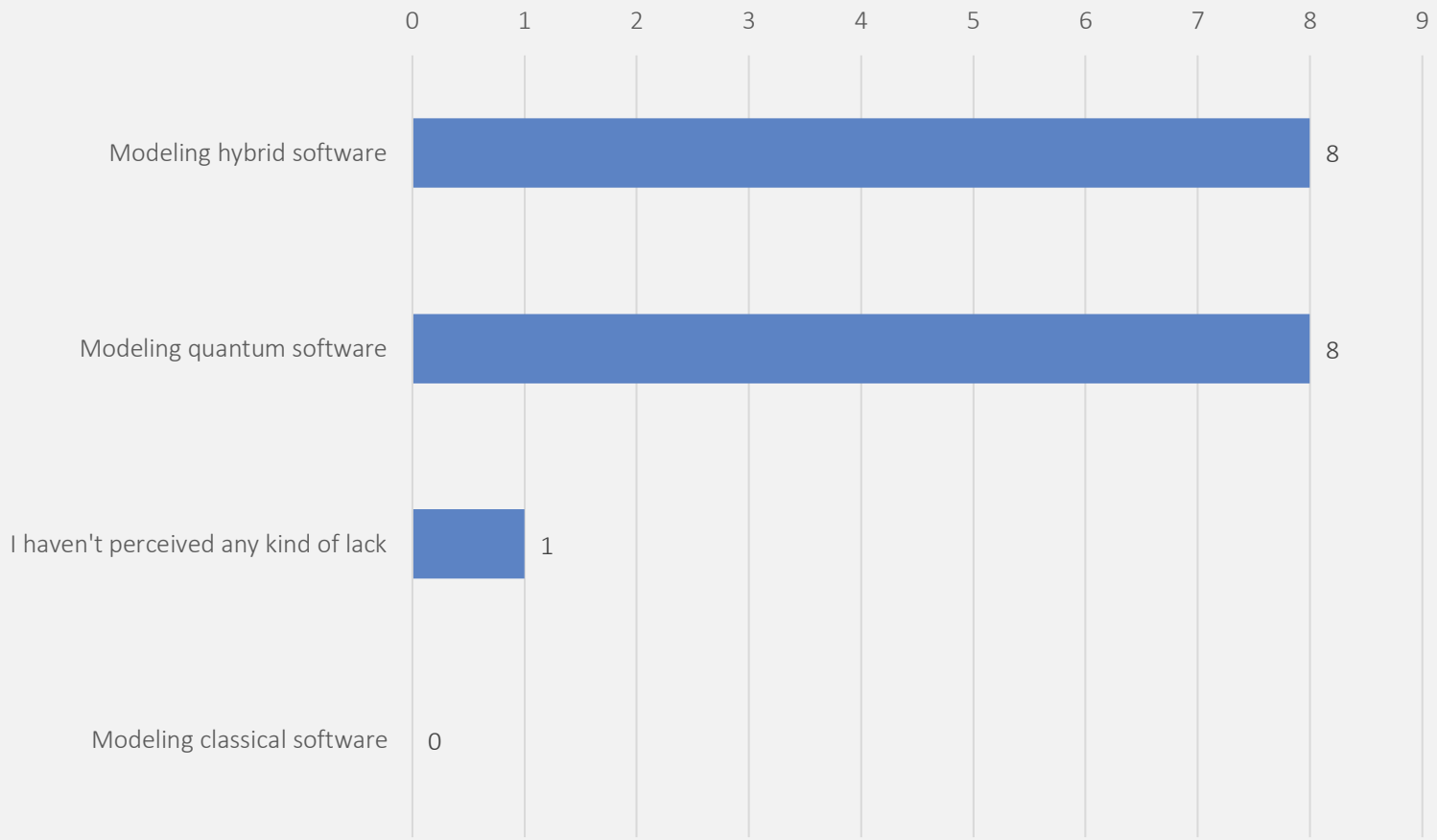
Lifecycle phases where models have been used, created or updated



Satisfaction degree with the employed modeling languages



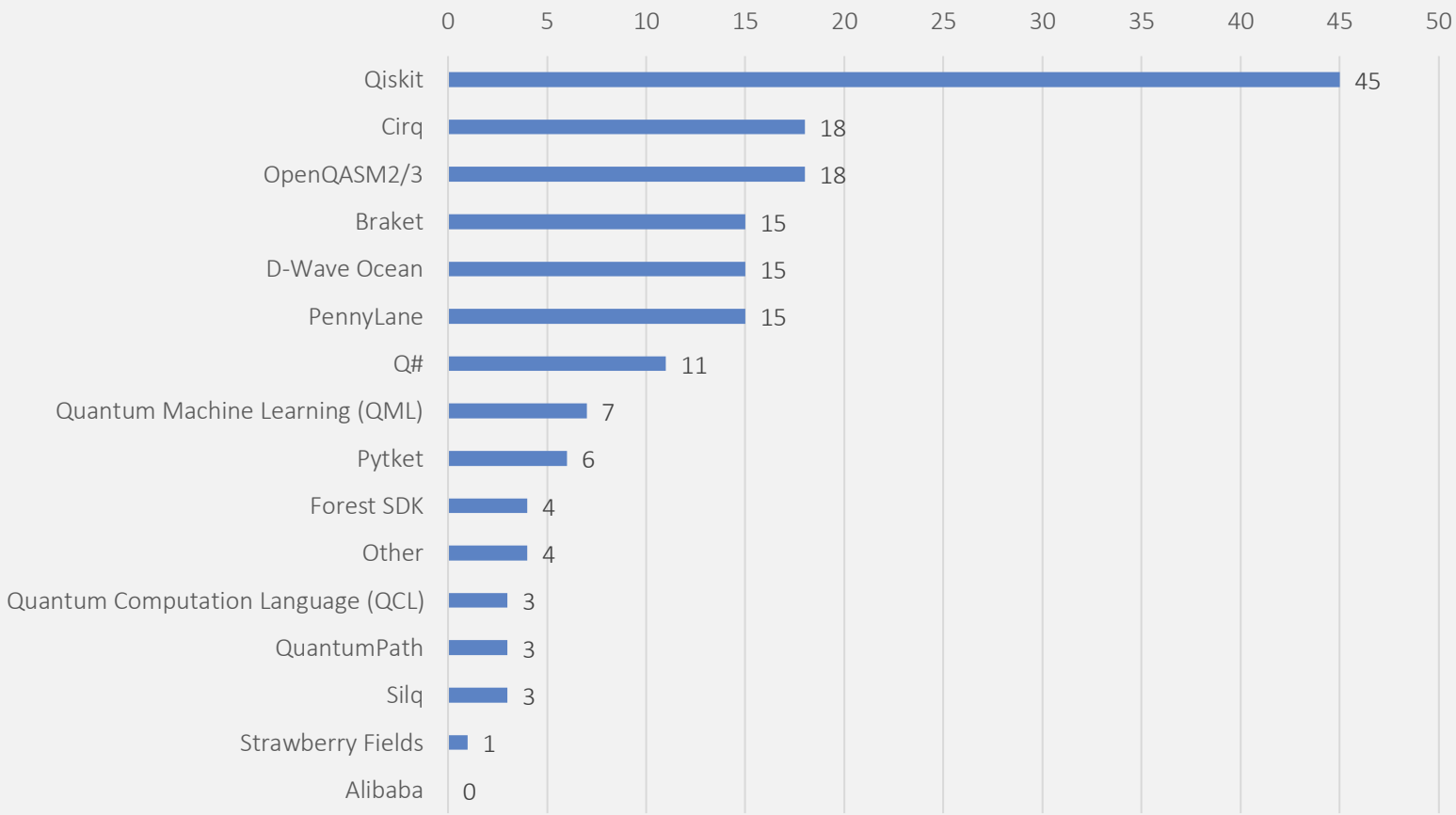
Purposes where a lack of modeling has been perceived



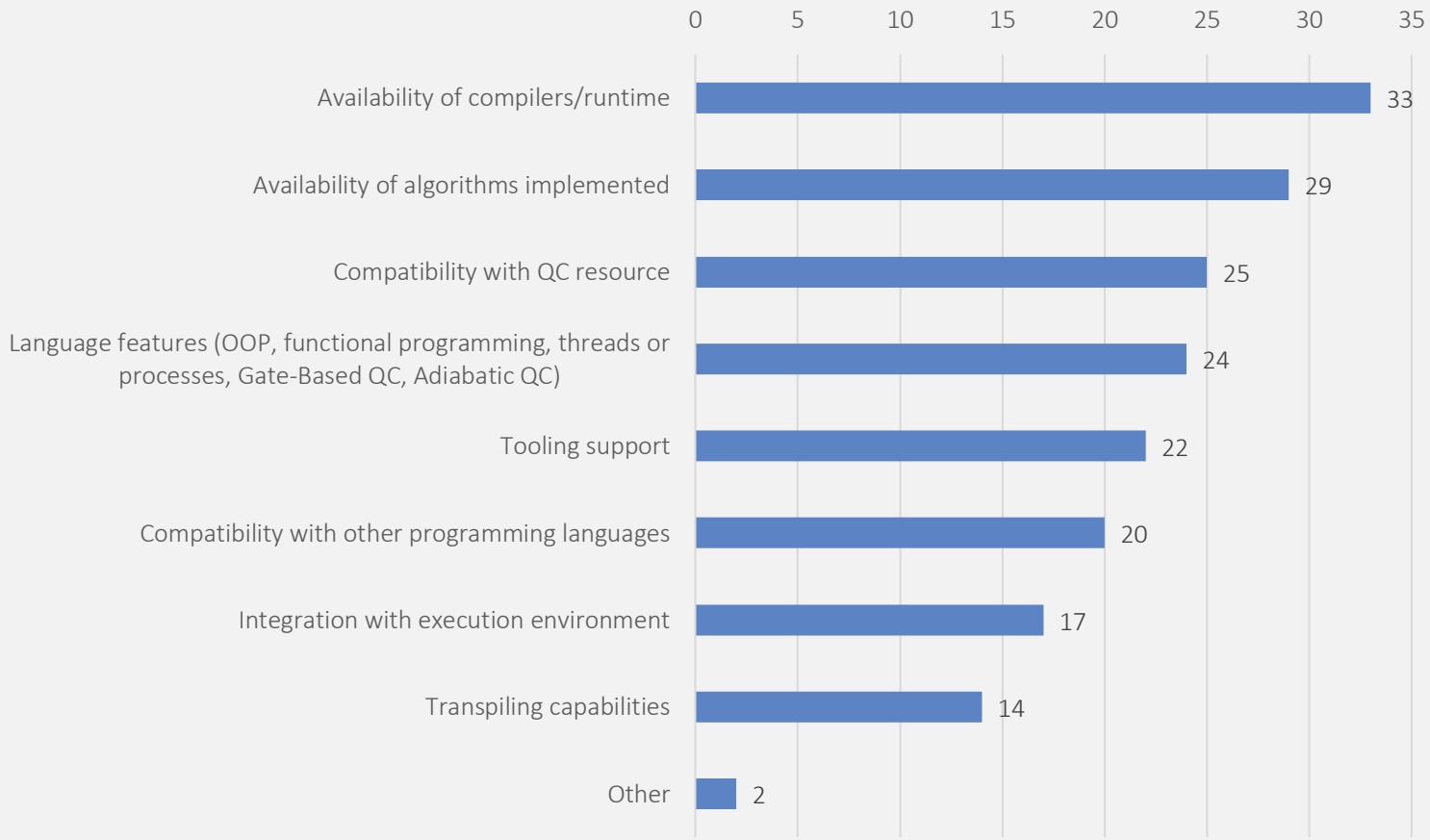
Results obtained from the survey

Goal 2: Implementation of quantum software

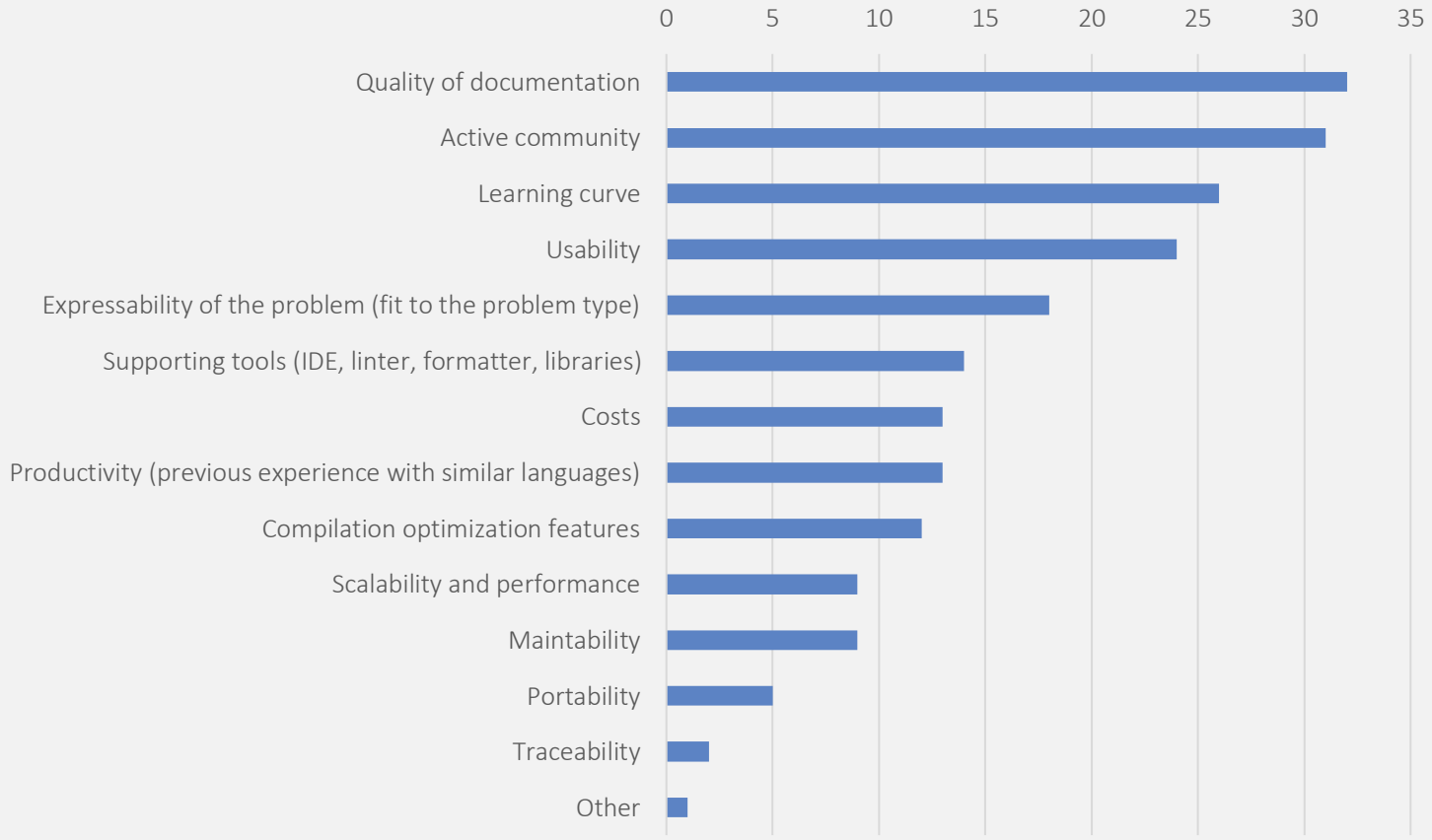
Quantum programming languages used



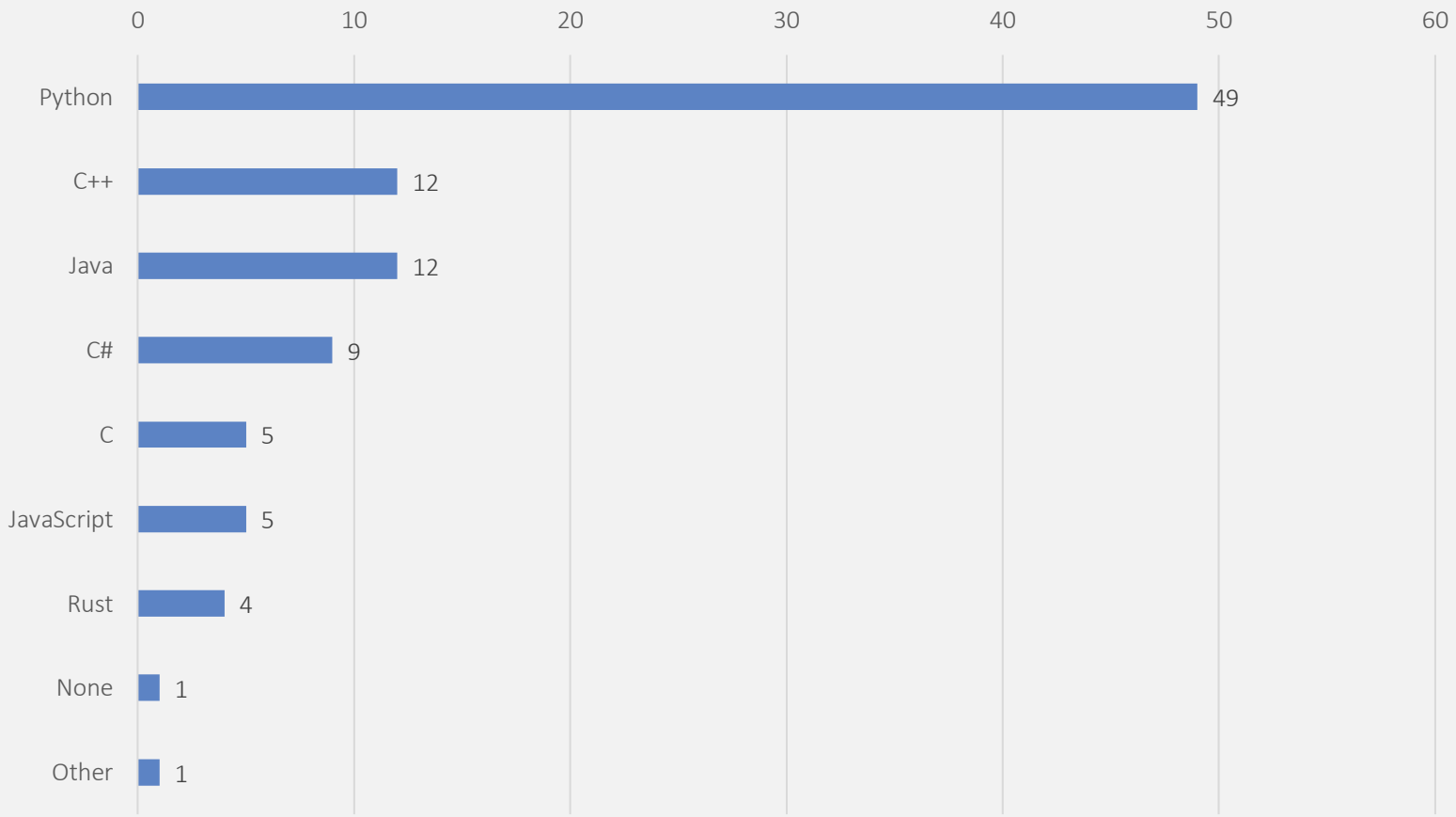
Reasons for choosing these quantum programming languages or toolkits (regarding functional requirements)



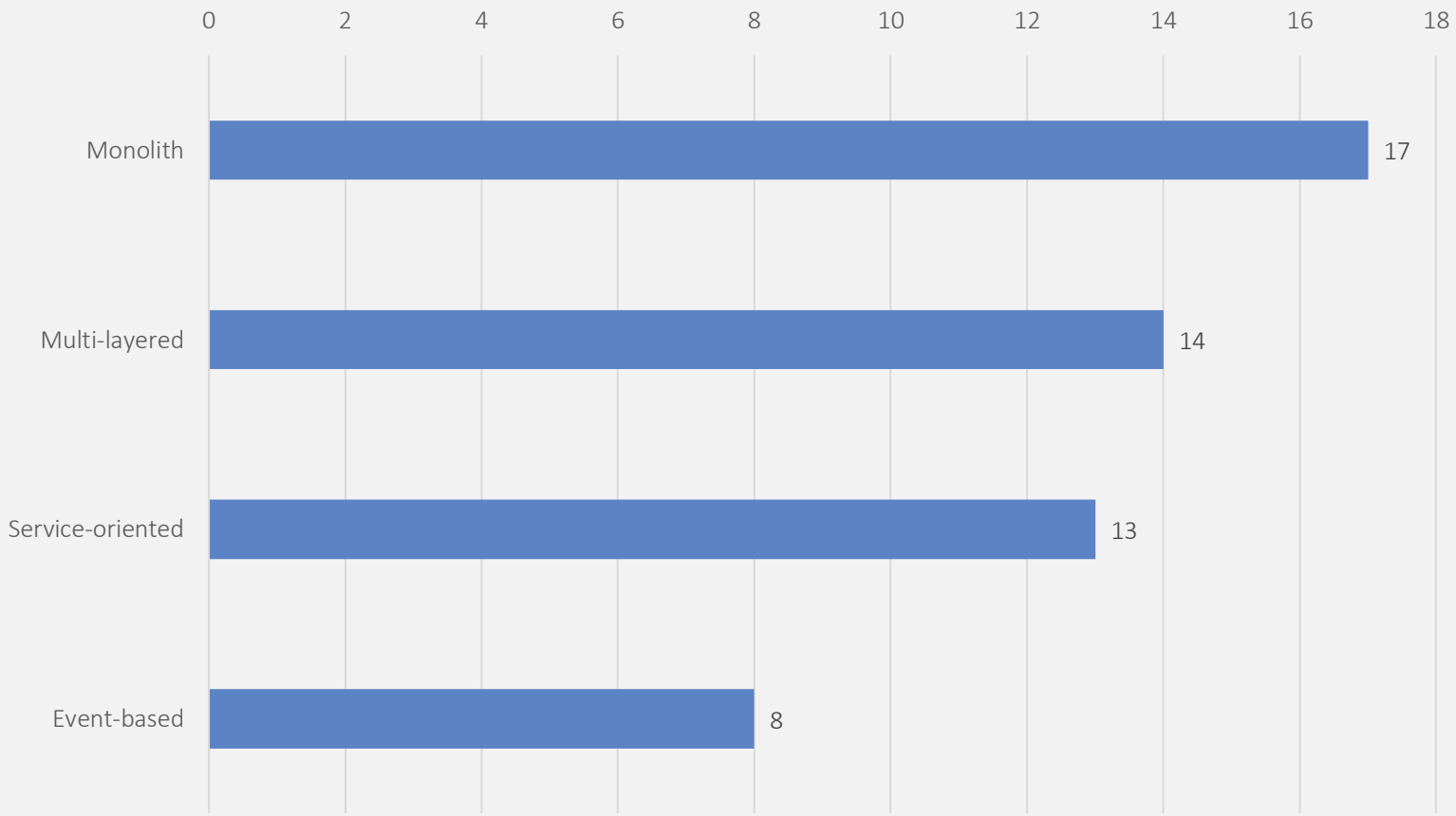
Reasons for choosing these quantum programming languages or toolkits (regarding non-functional requirements)



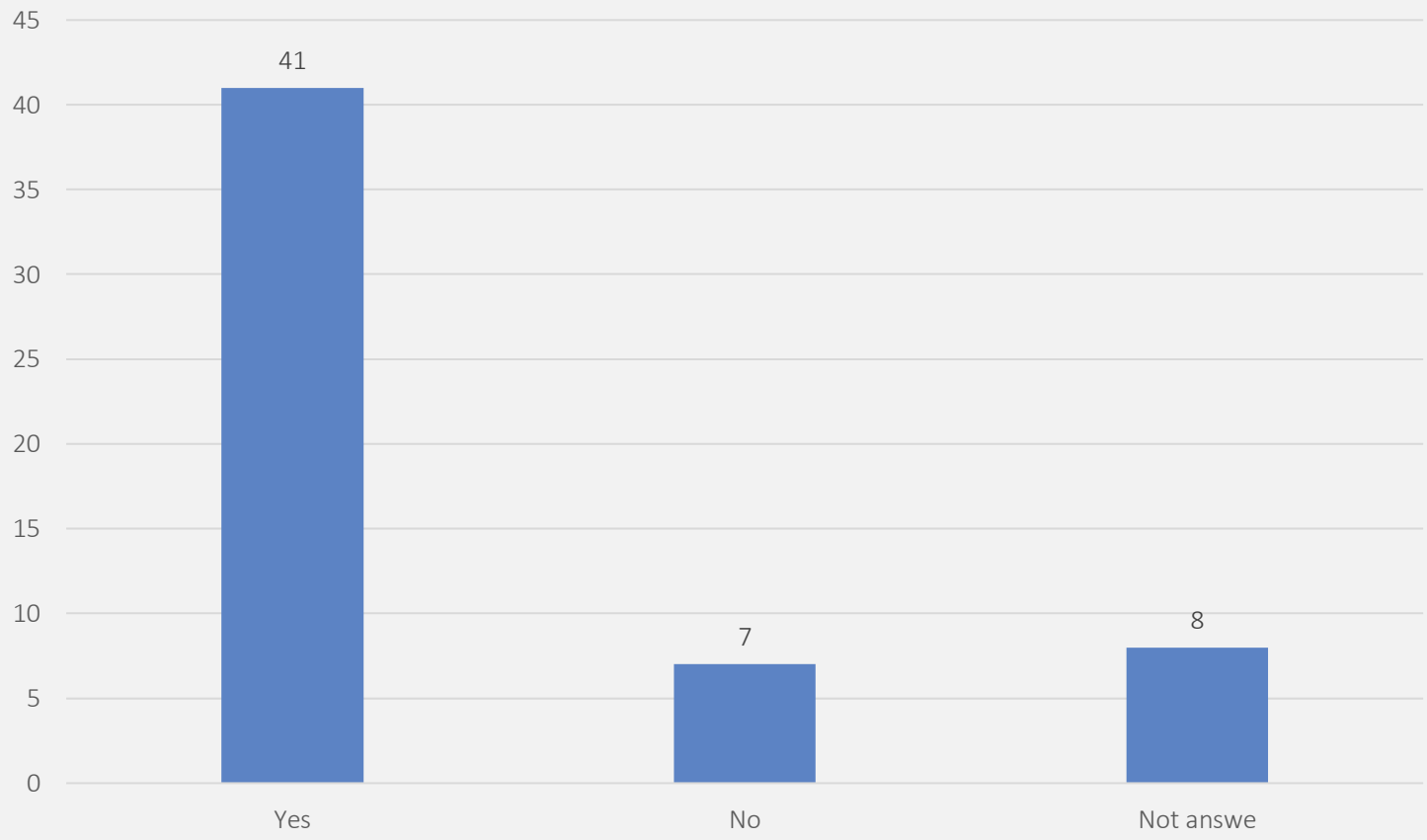
Classical programming languages used for developing hybrid software



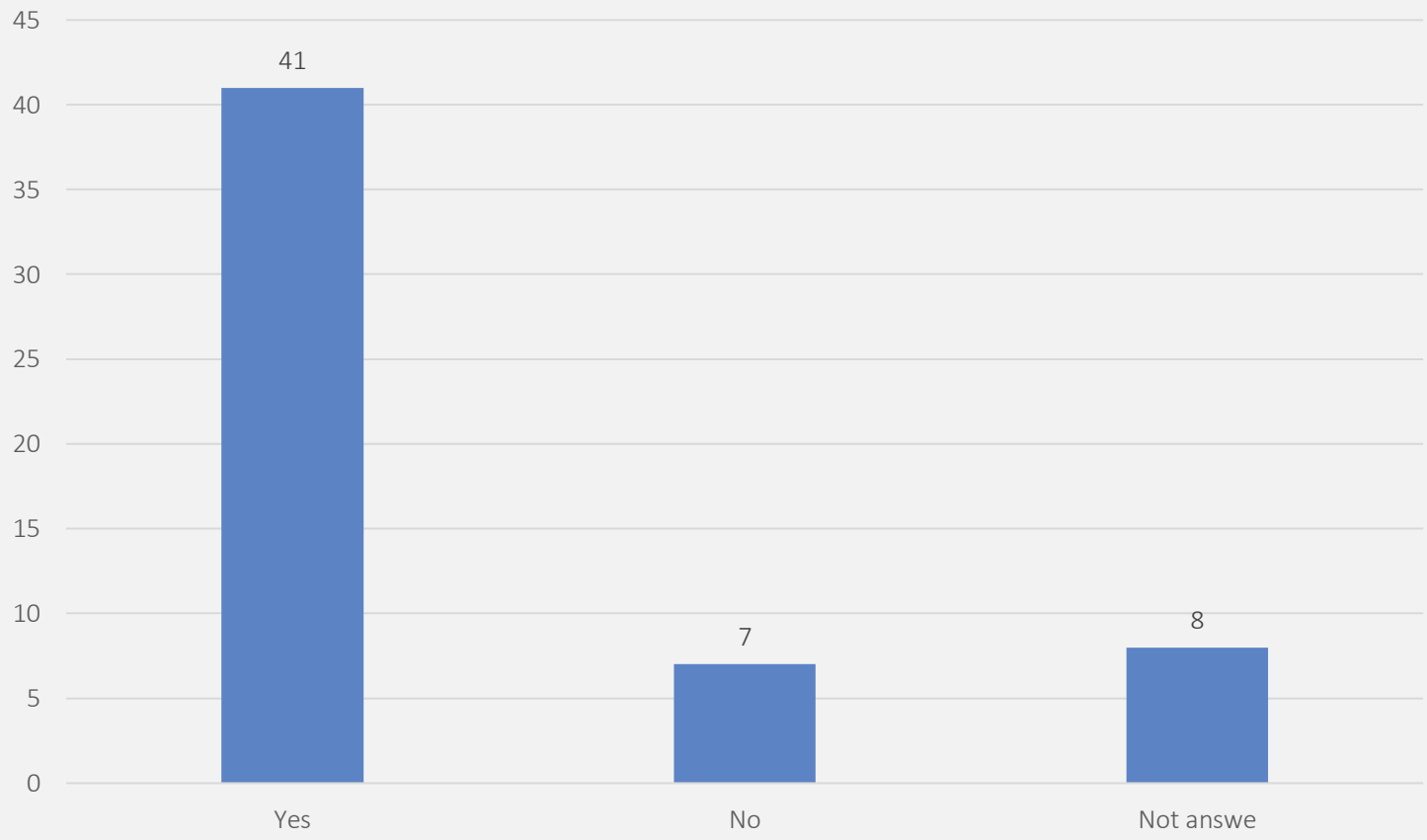
Type of the architectures of the hybrid application systems



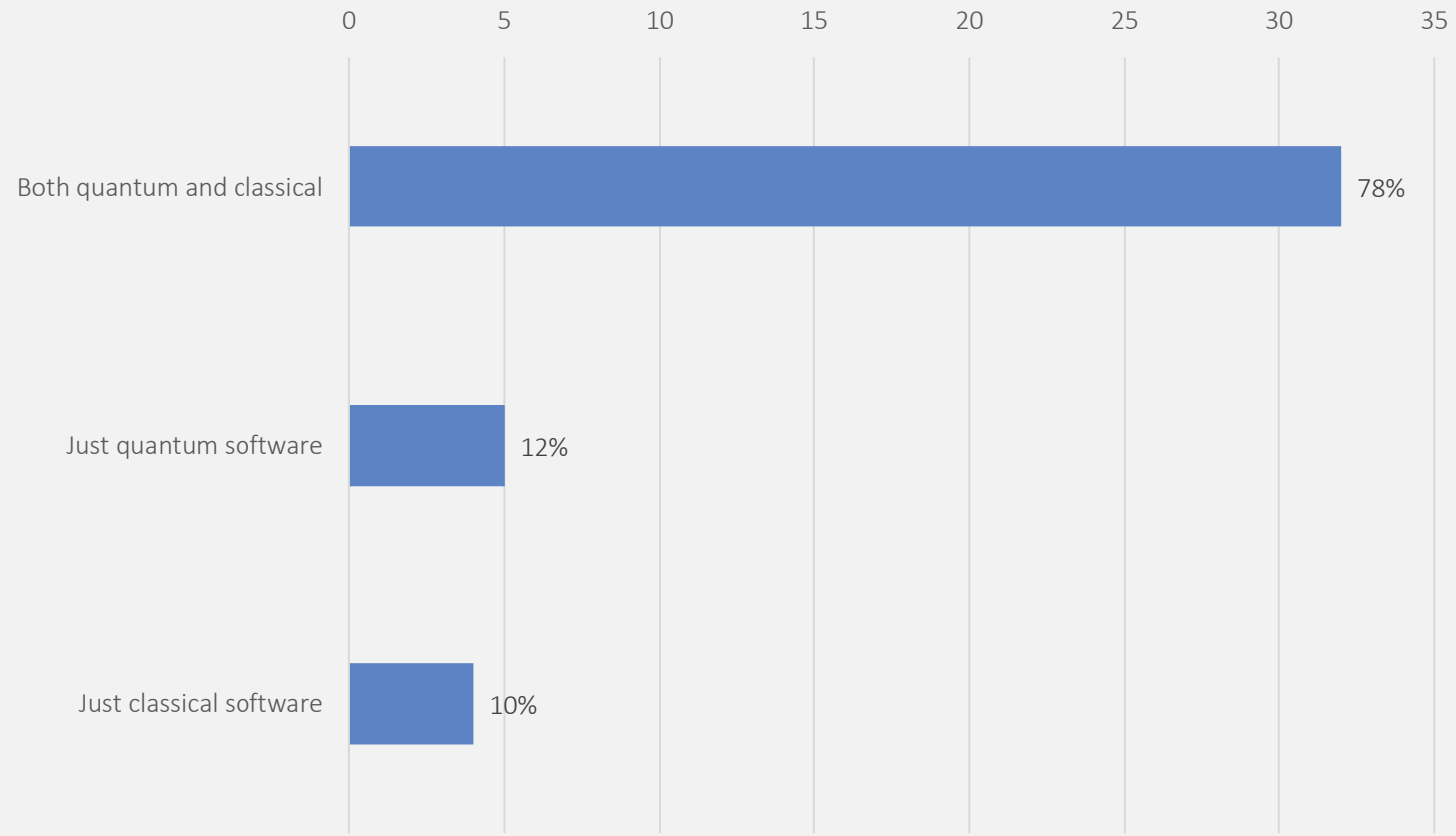
Is hybrid software tested?



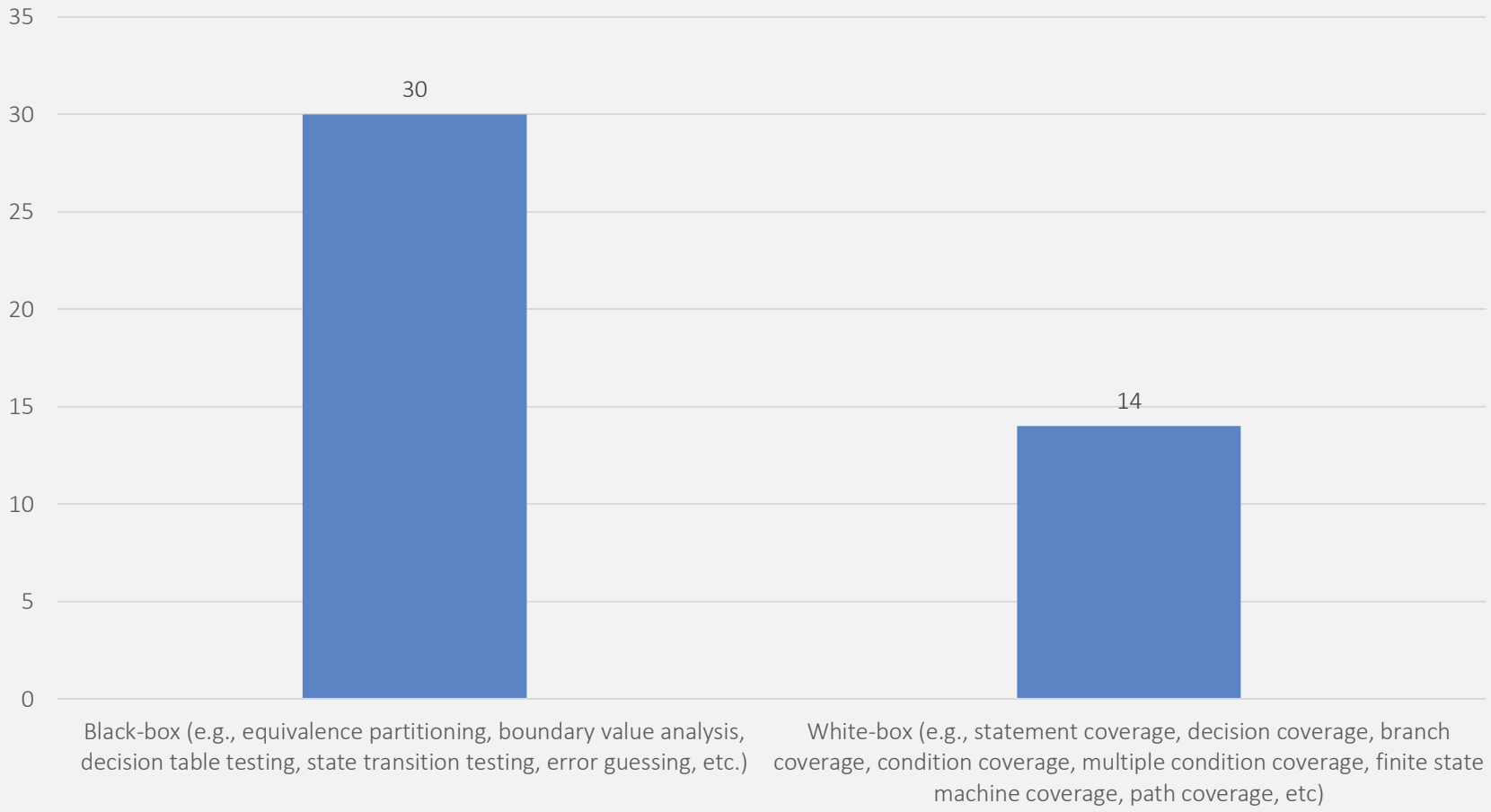
Is hybrid software tested?



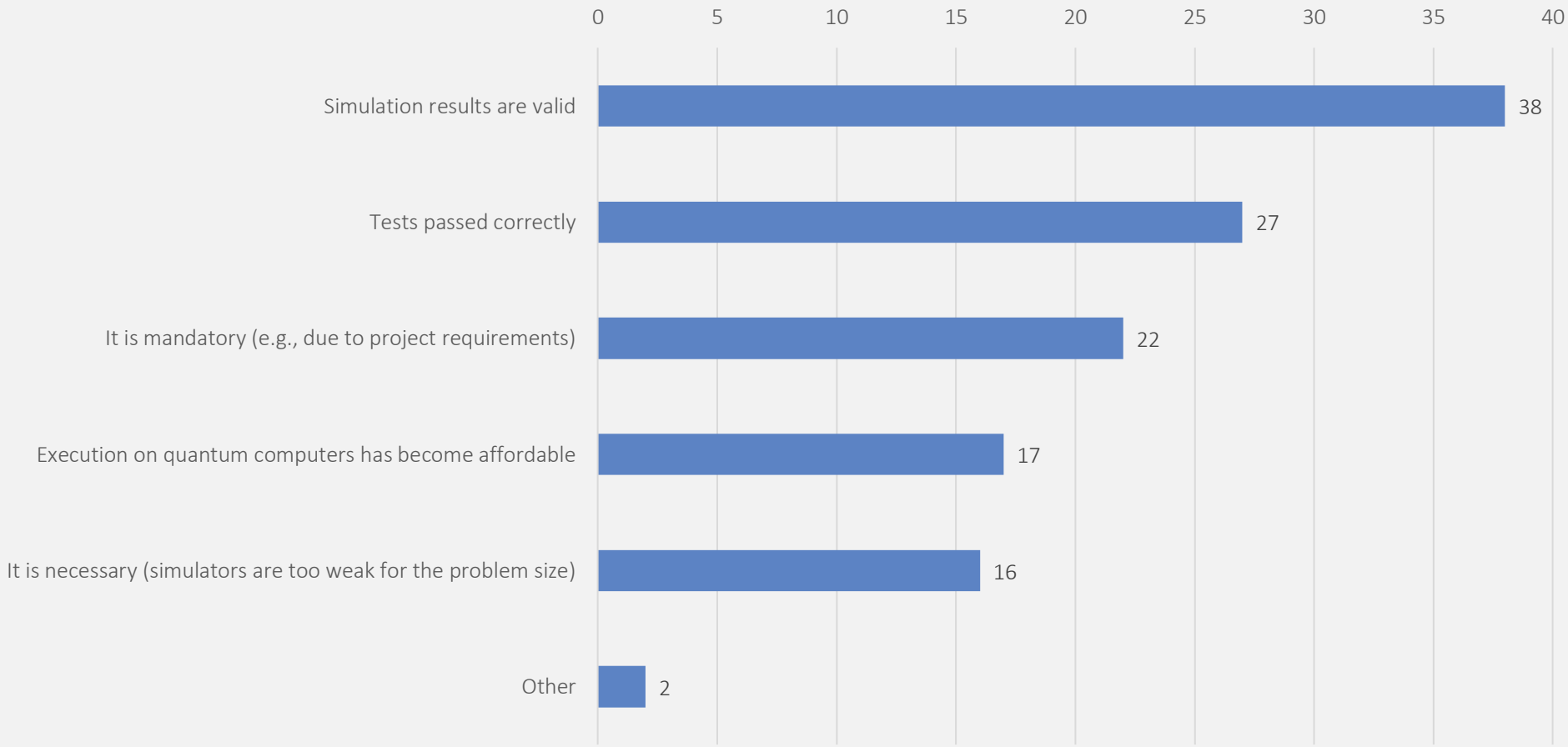
Software components tested



Testing techniques followed



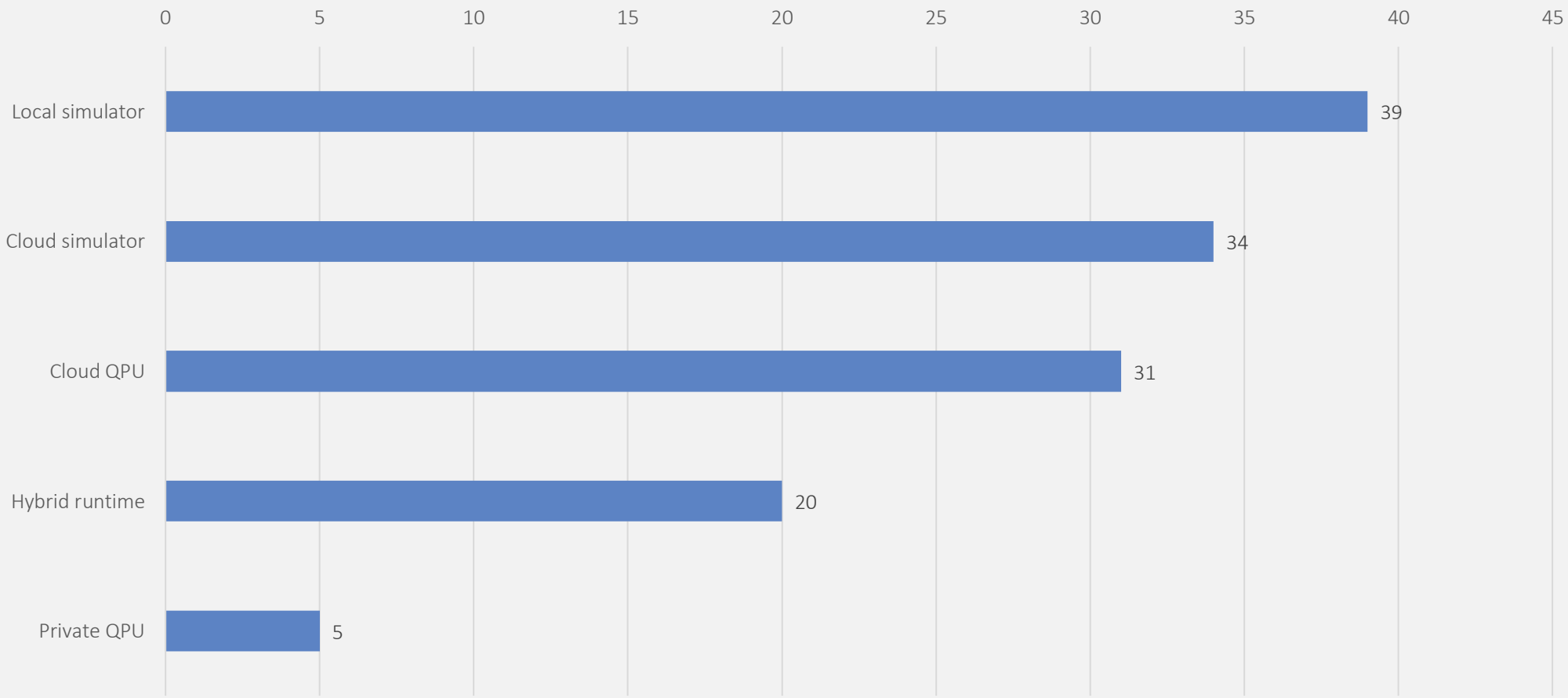
Criteria for moving from a simulation environment to a real quantum computer



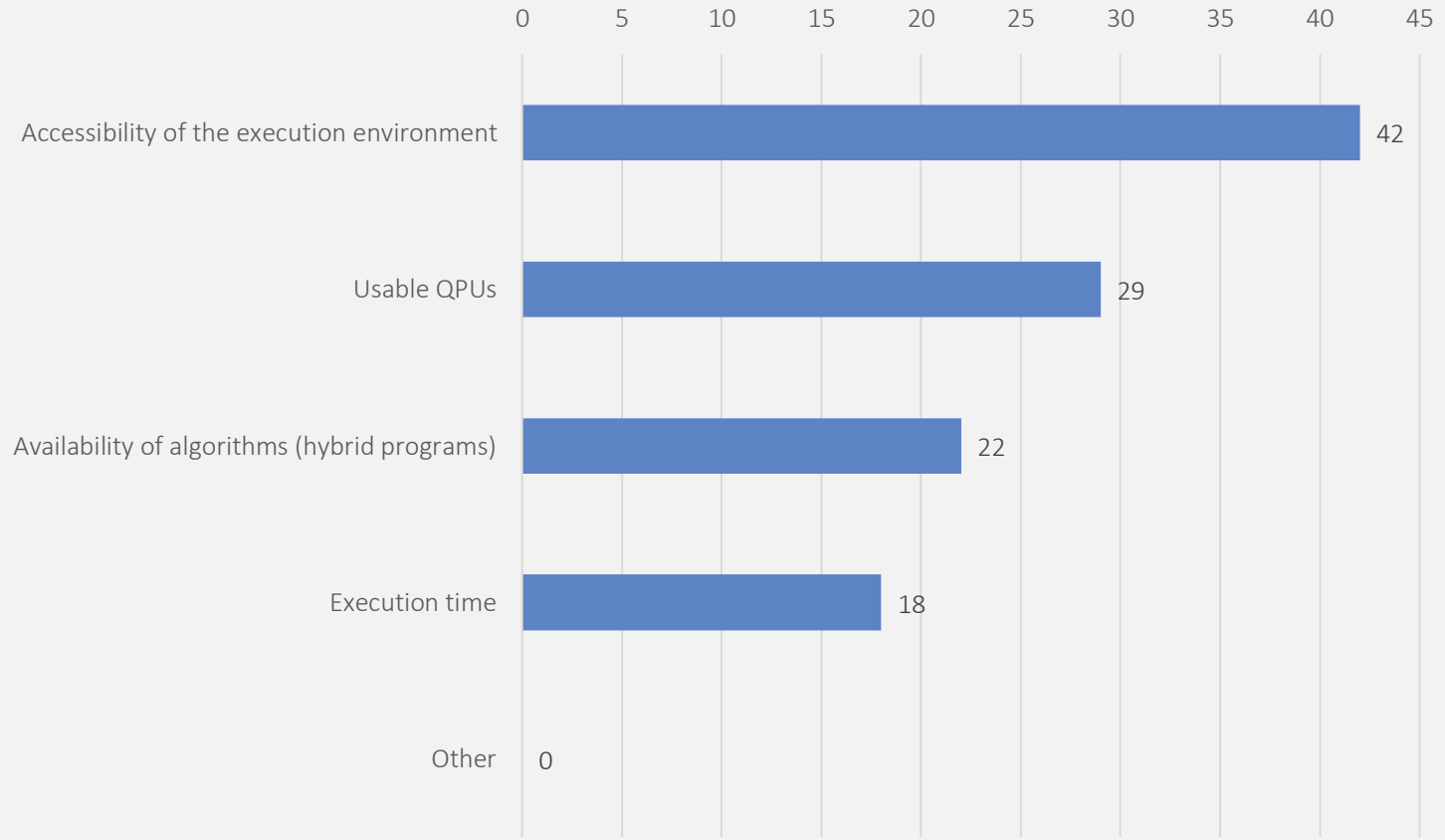
Results obtained from the survey

Goal 3: Operation of quantum software

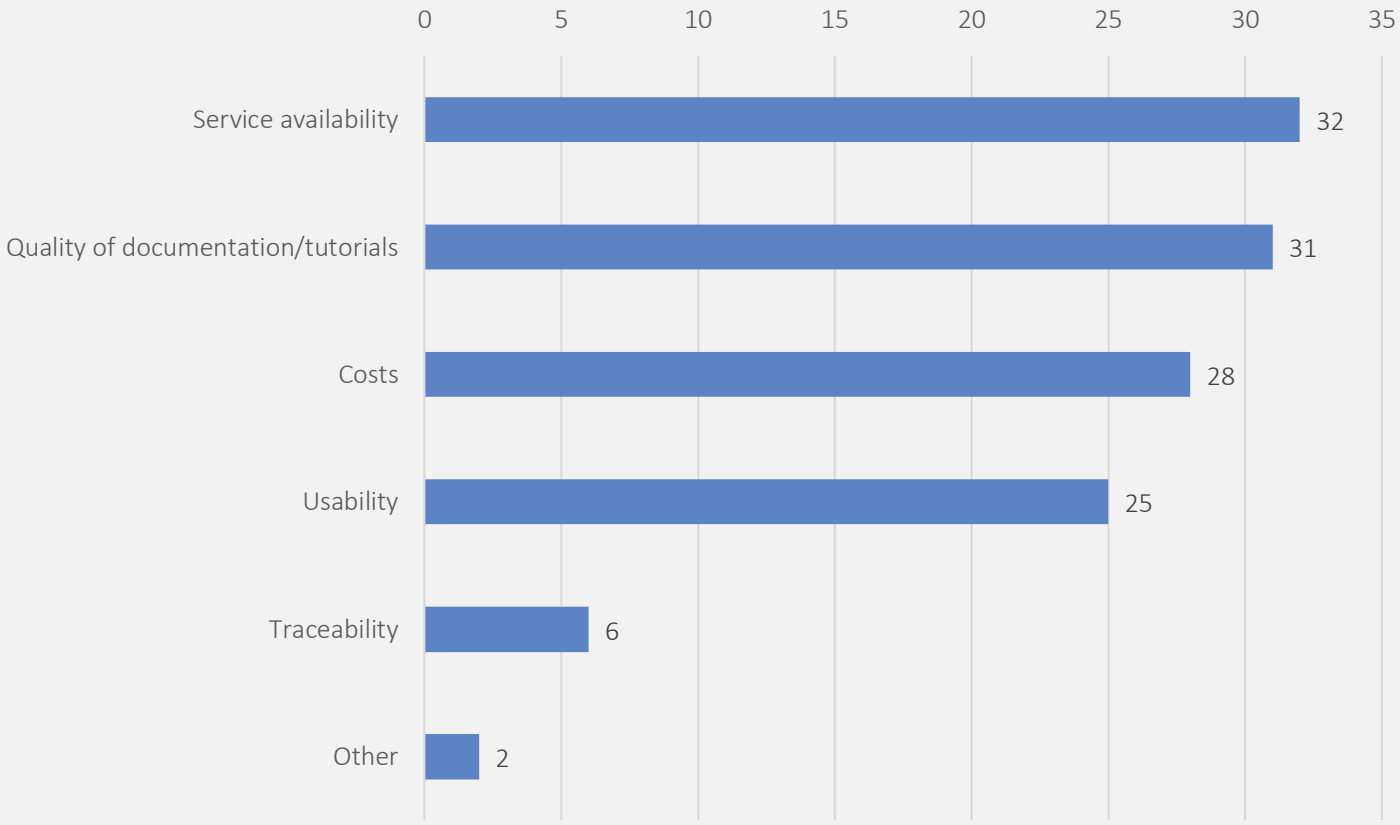
Execution environment used



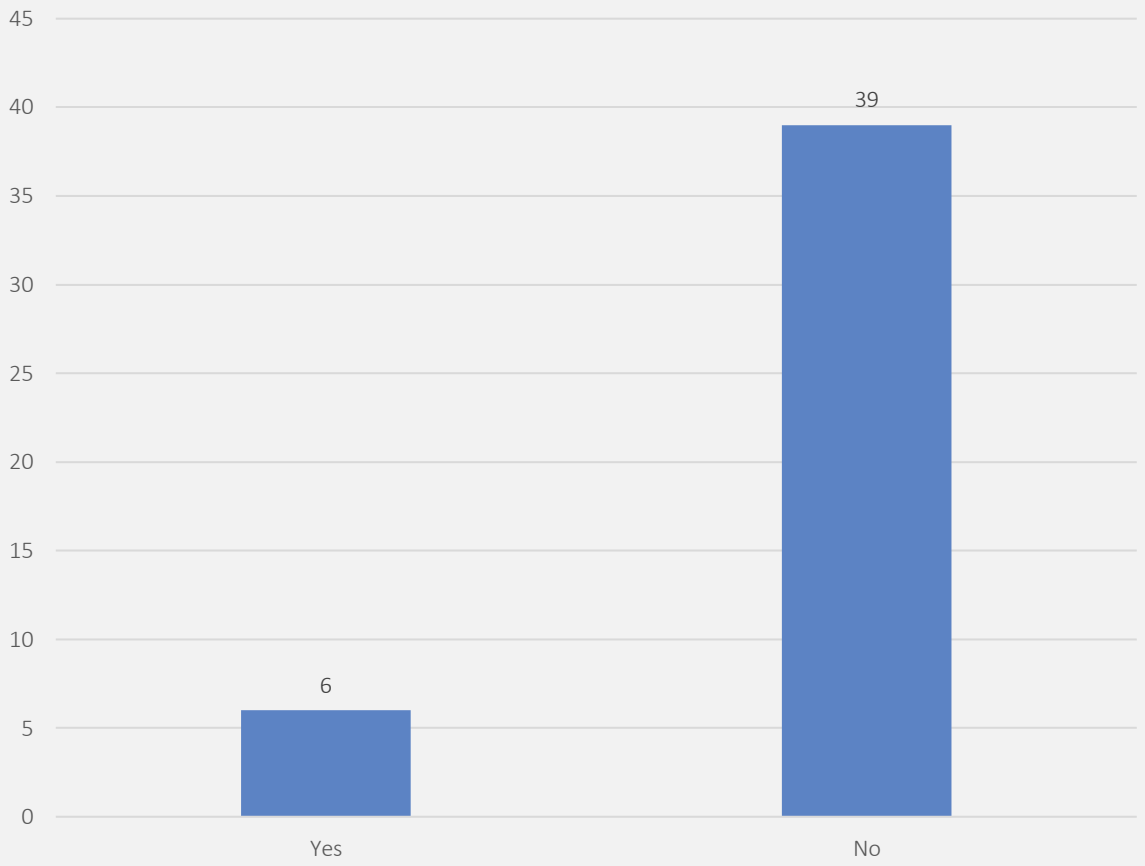
Reasons for choosing these execution environments (regarding functional requirements)



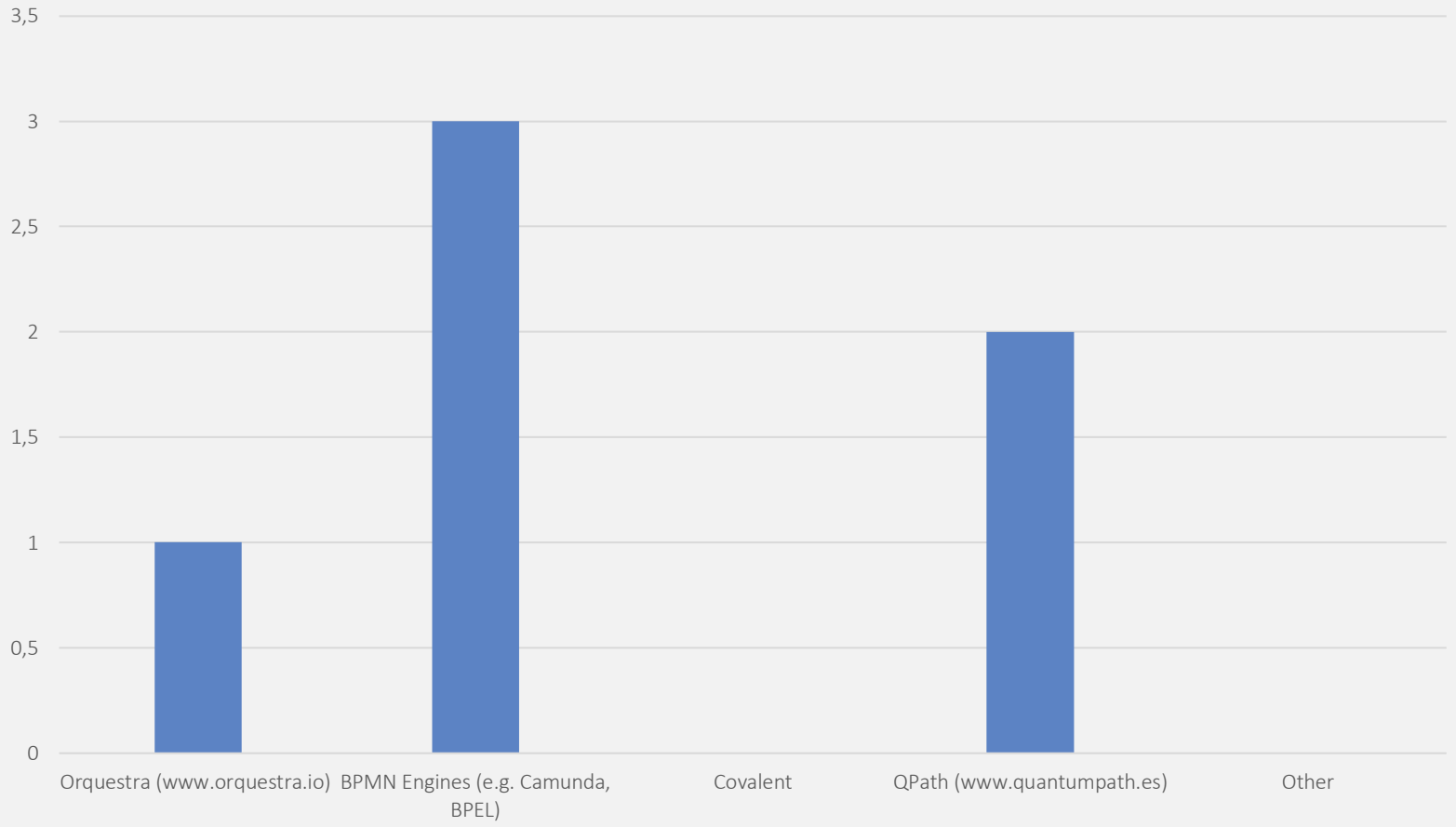
Reasons for choosing these execution environments (regarding non-functional requirements)



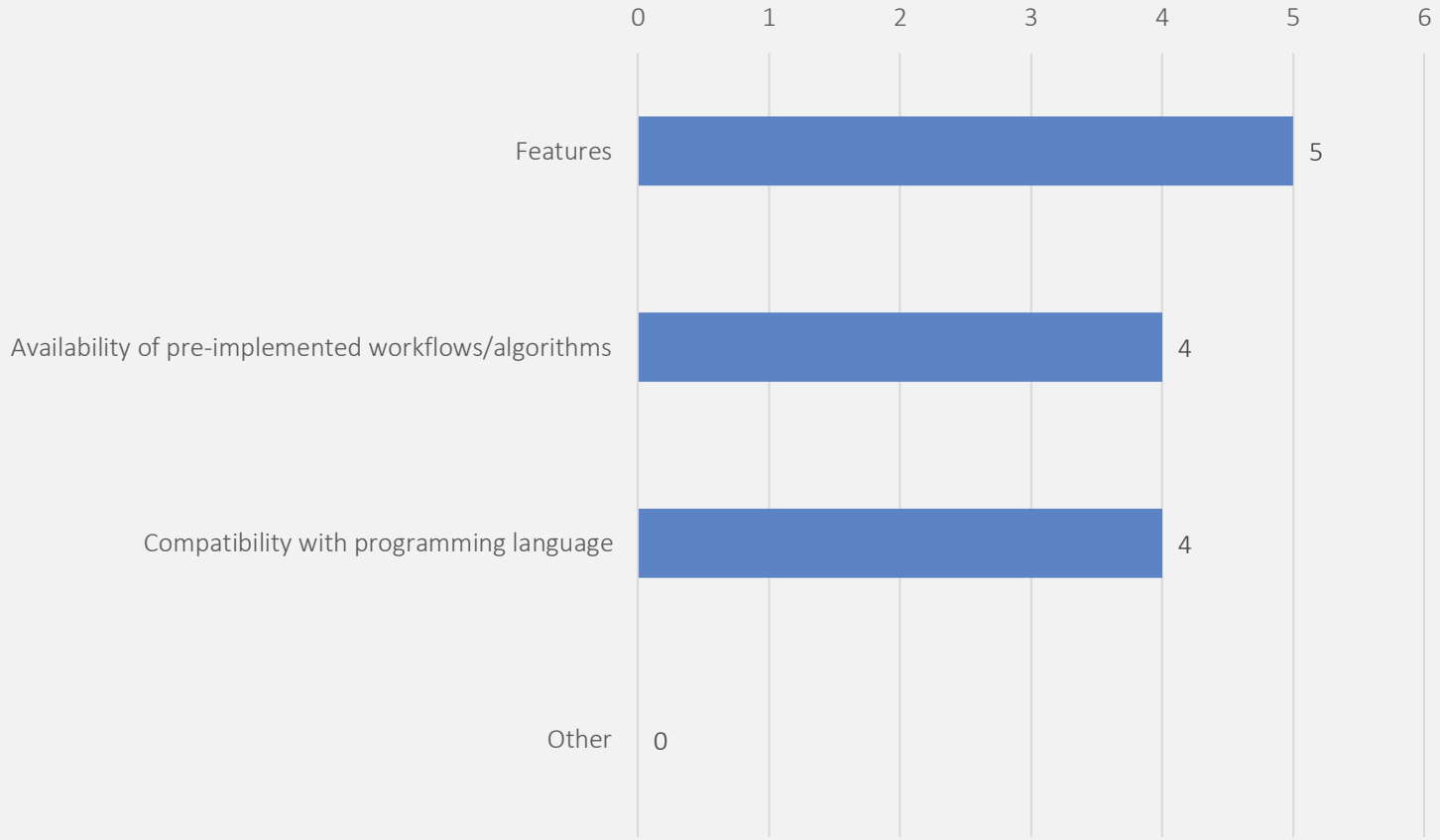
Use of workflow technologies



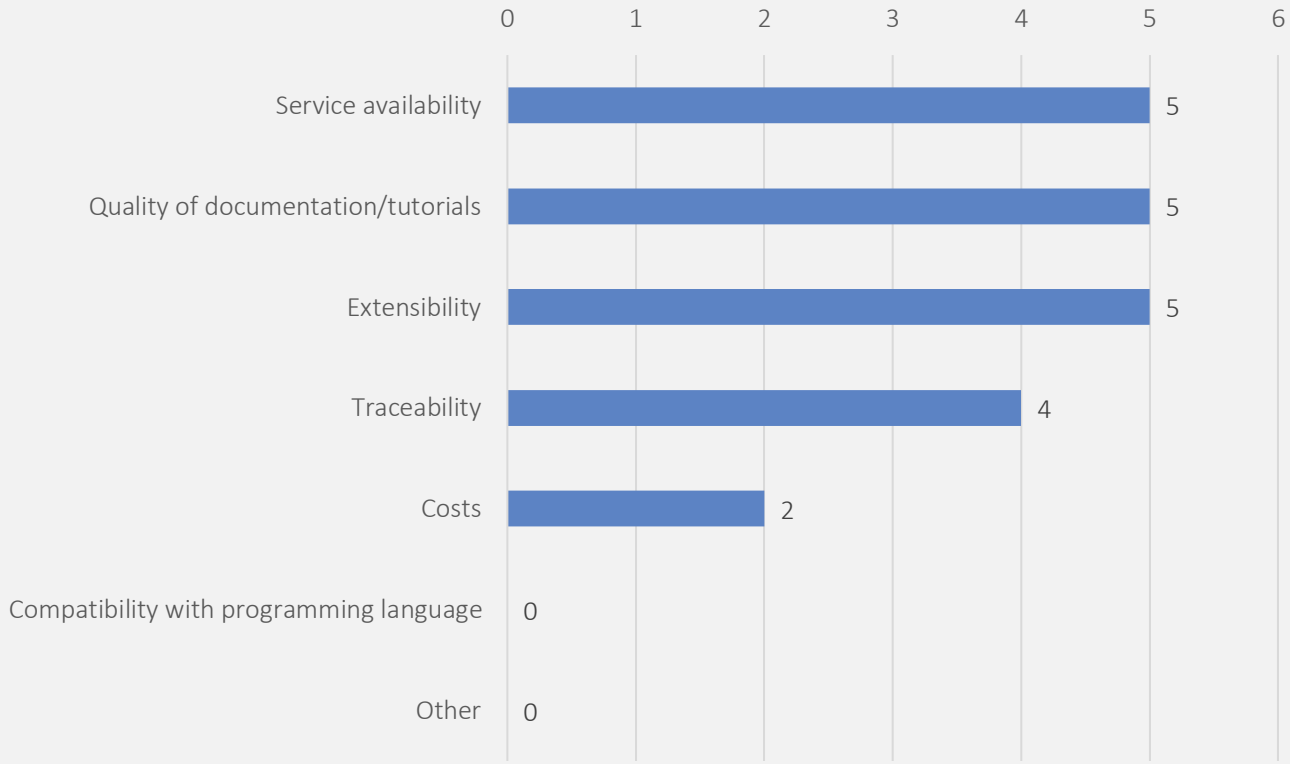
Workflow technologies used



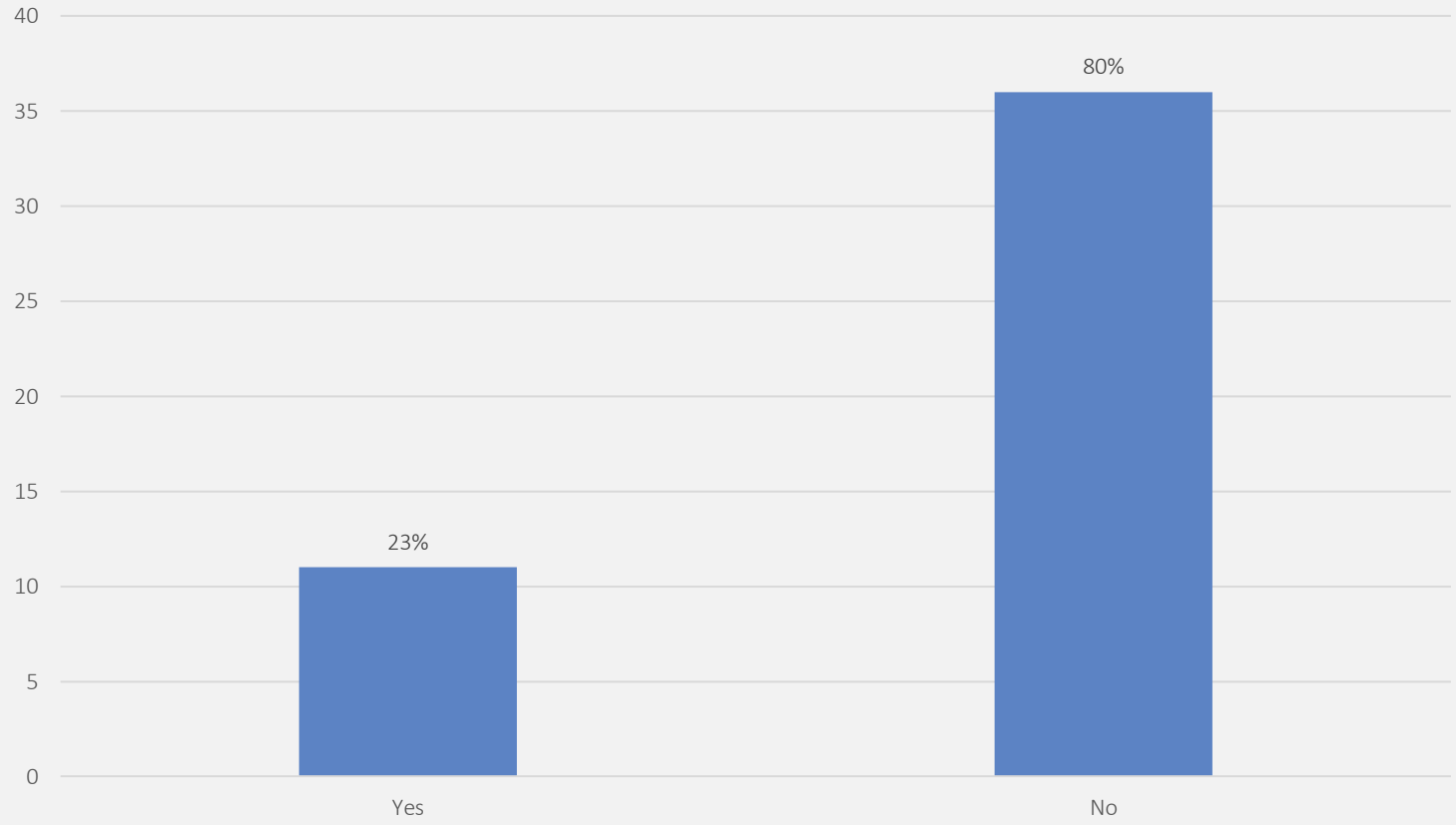
Reasons for choosing workflow technologies (regarding functional requirements)



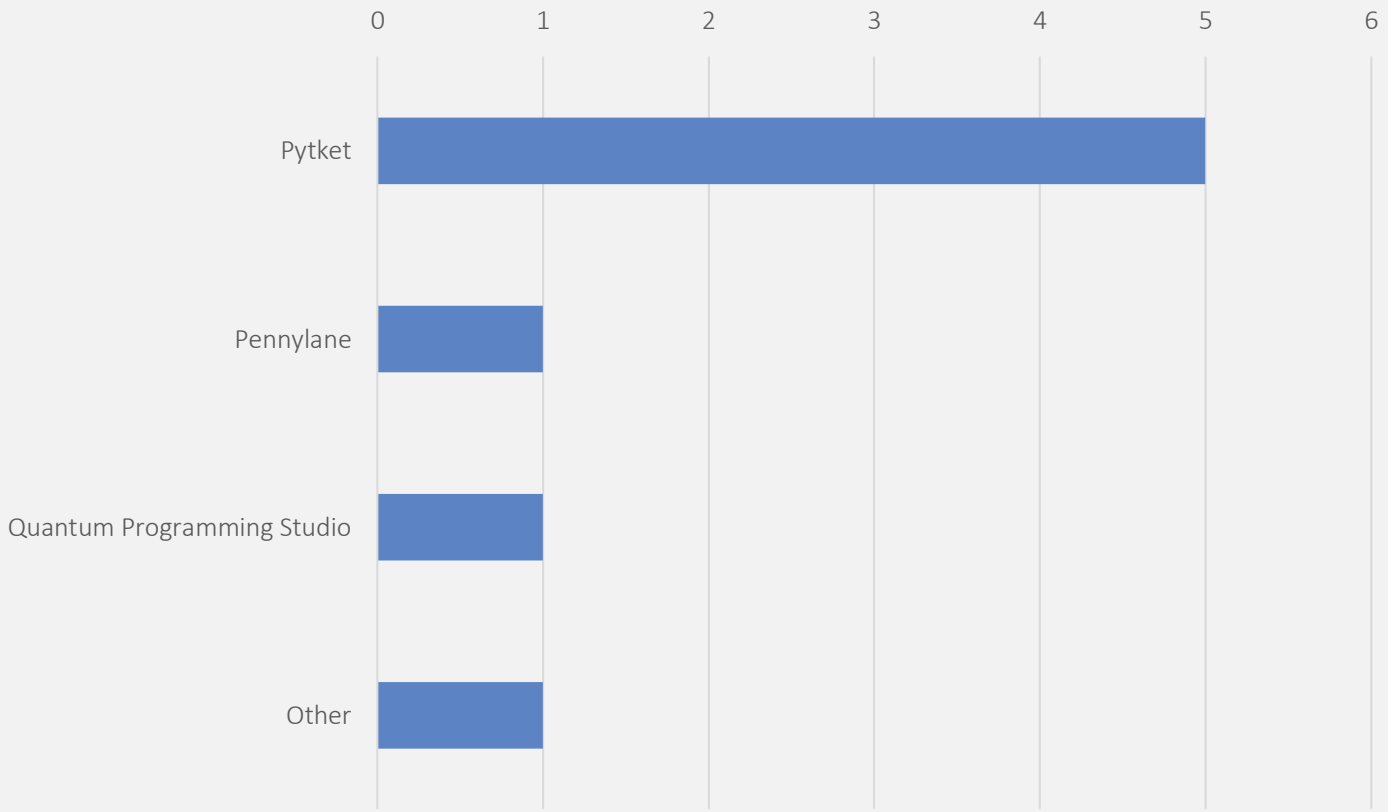
Reasons for choosing workflow technologies (regarding non-functional requirements)



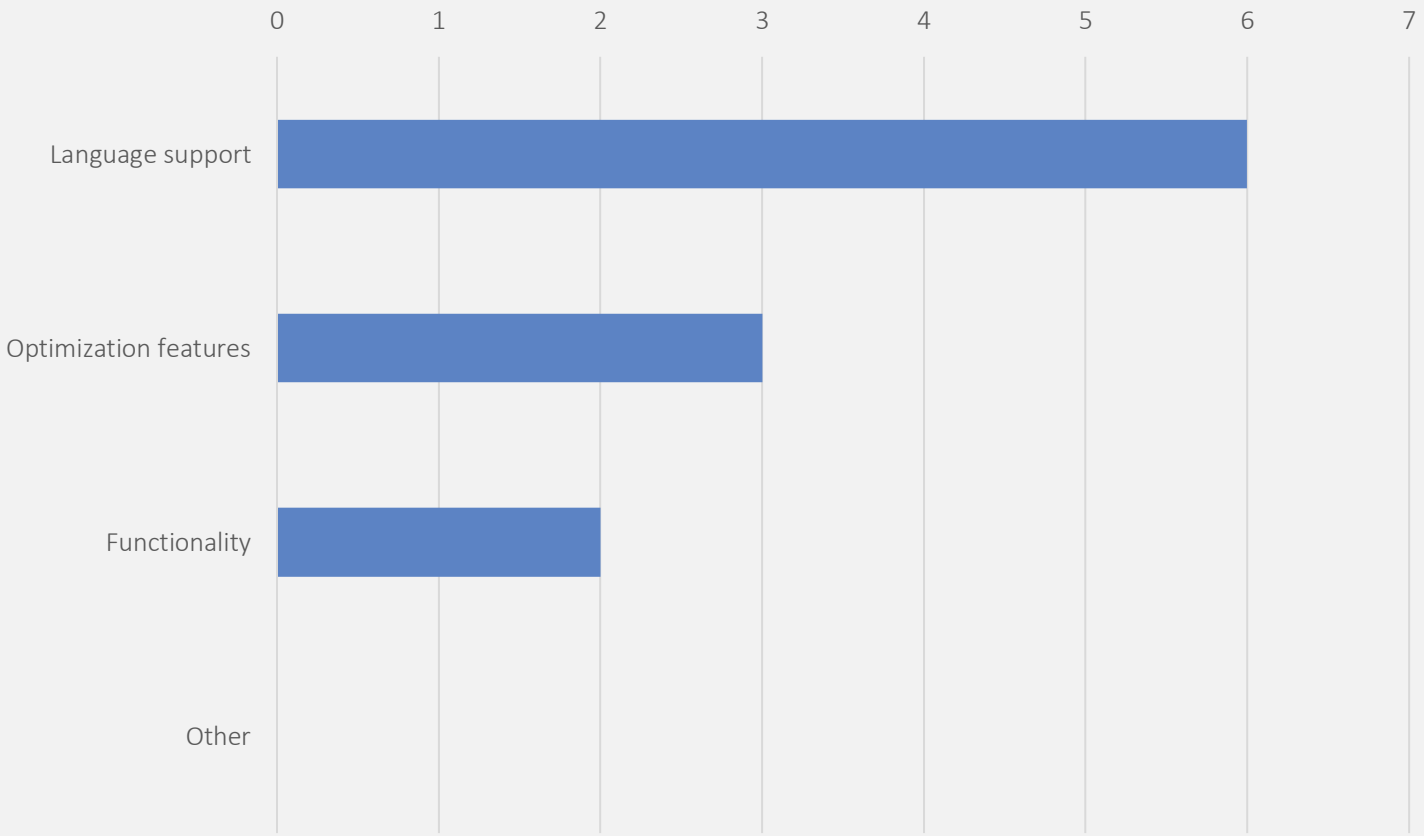
Use of language converters



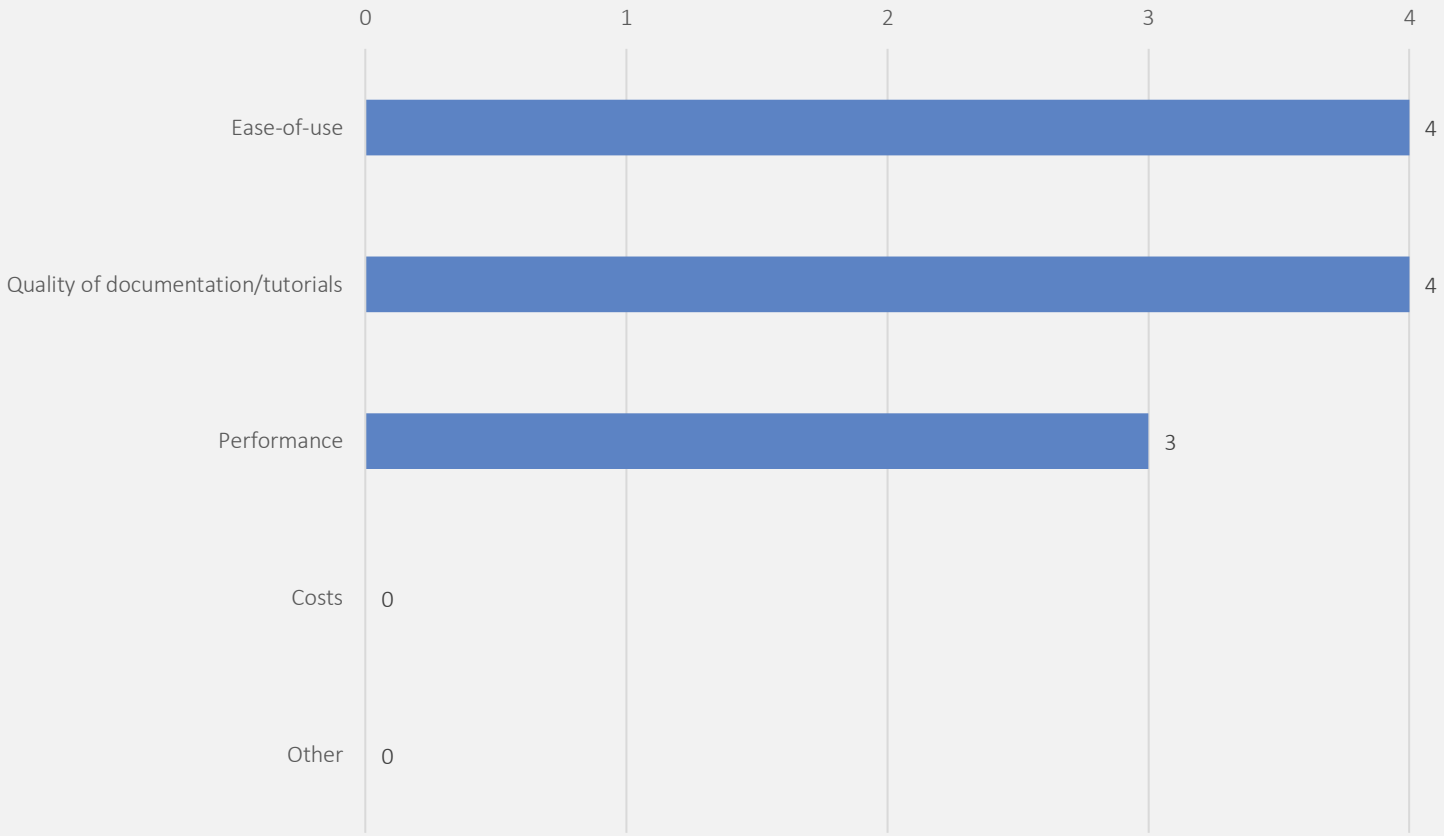
Standalone language converters employed



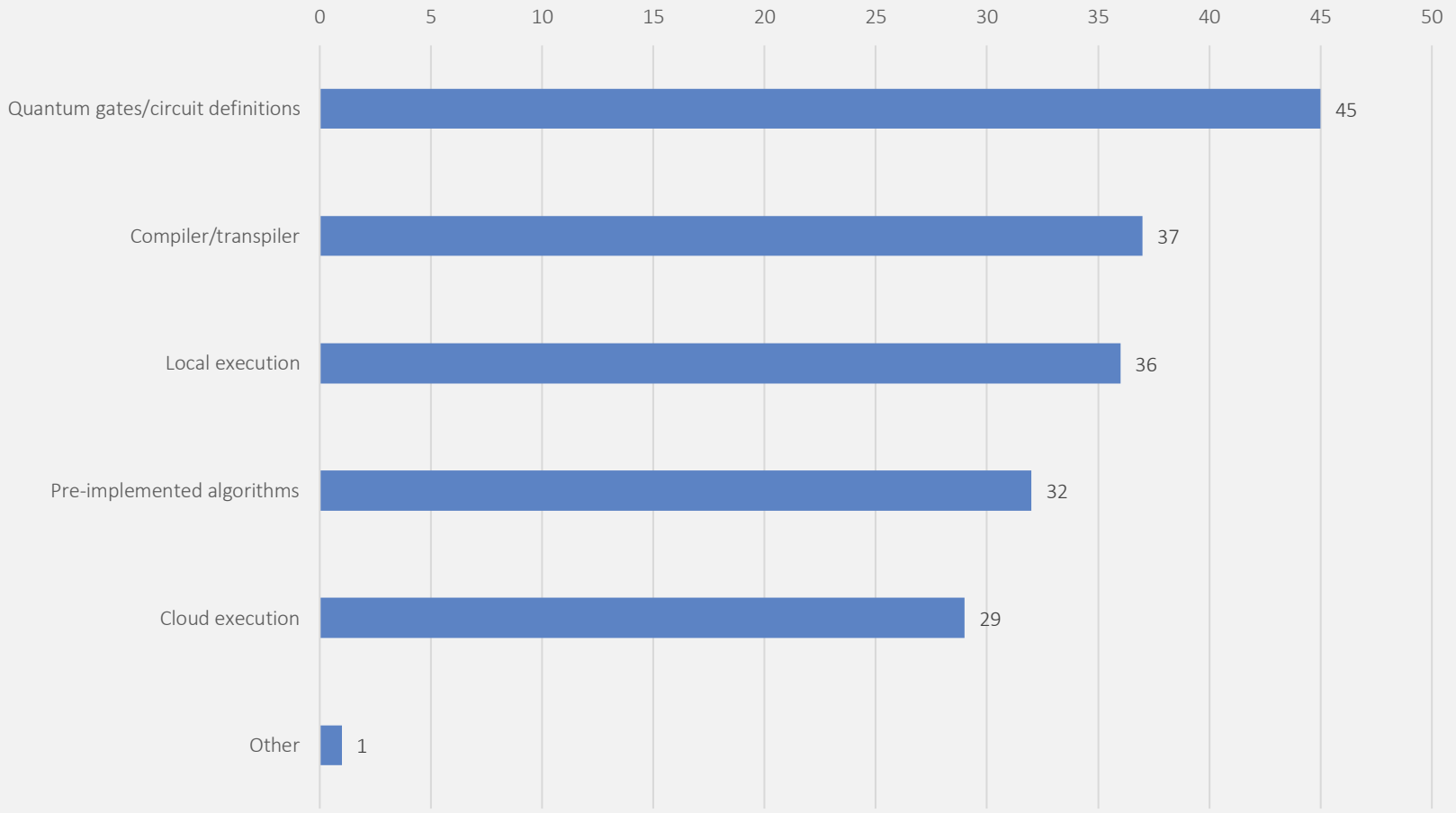
Reasons for selecting these language converters (regarding functional requirements)



Reasons for selecting these language converters (regarding non-functional requirements)



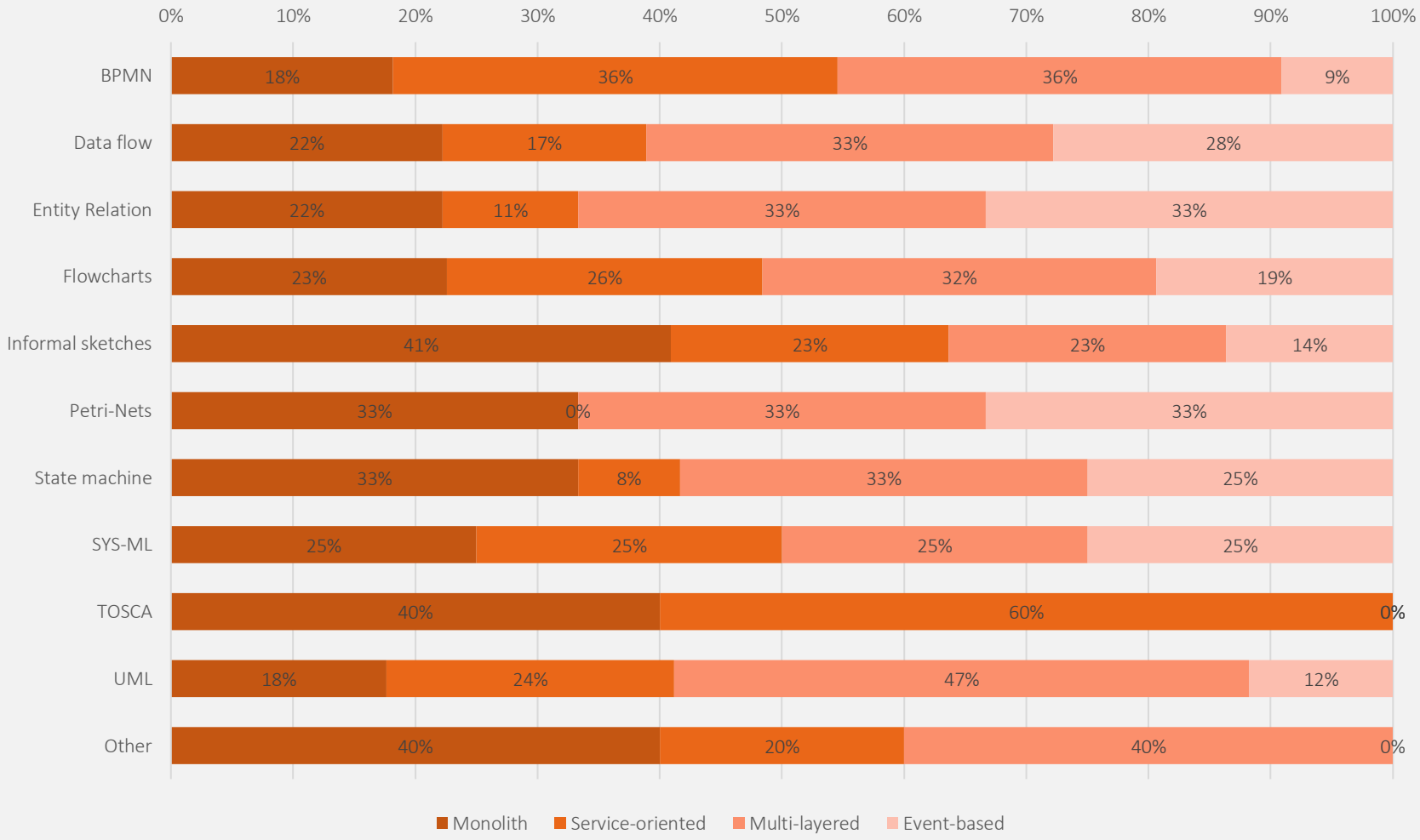
Functionalities used of the SDKs



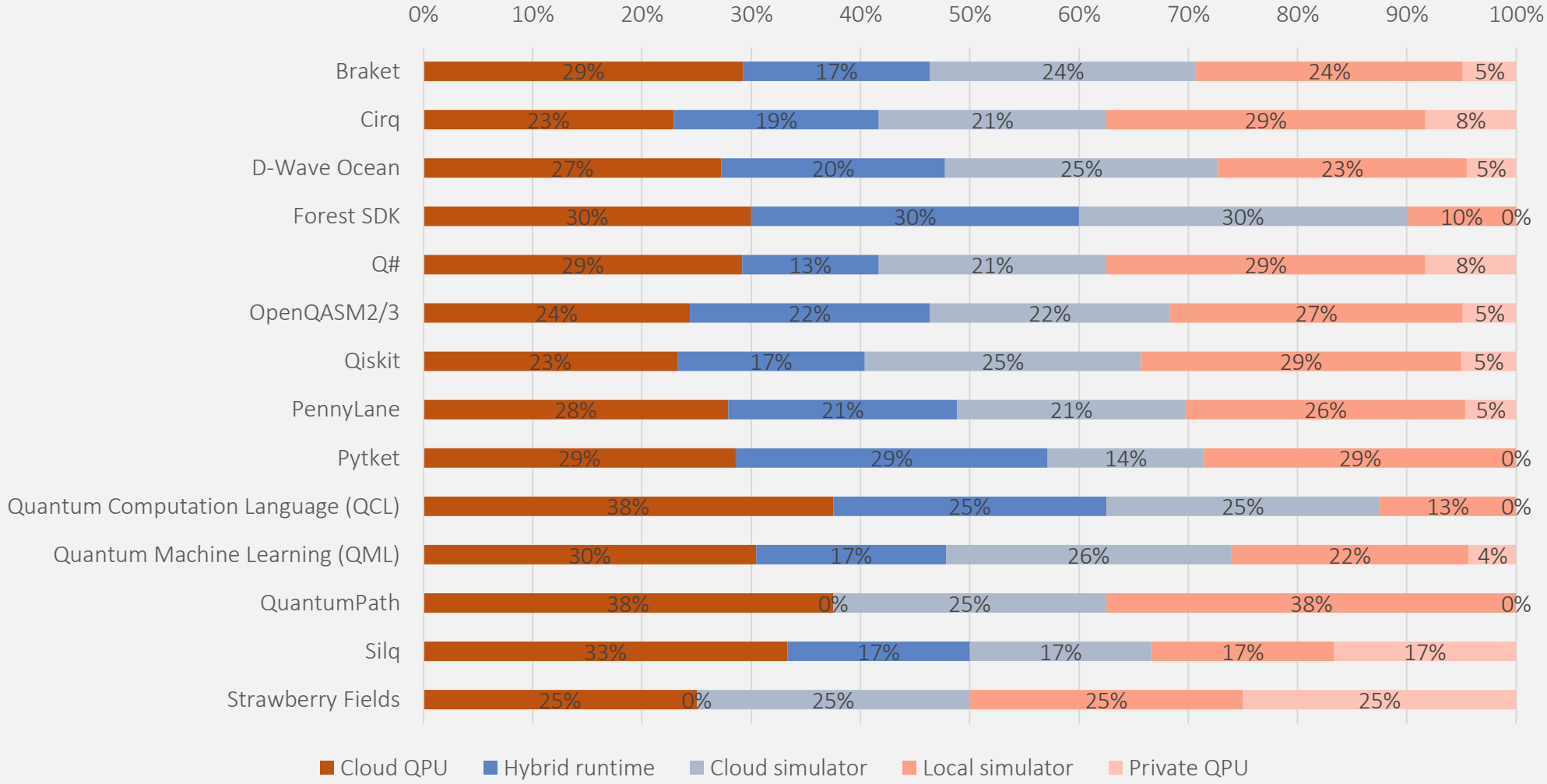
Results obtained from the survey

Cross-goals insights

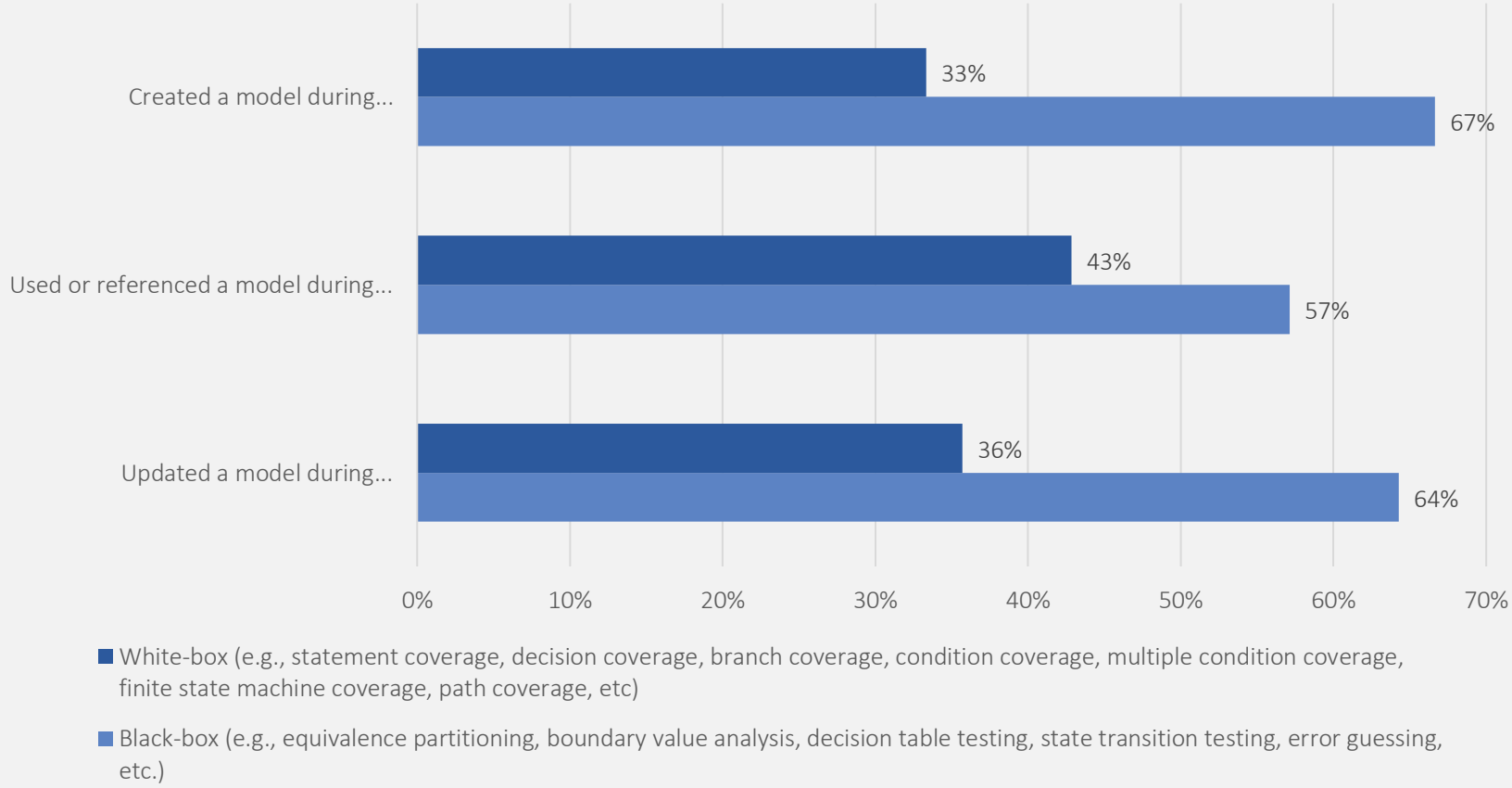
Architecture followed within each modeling language

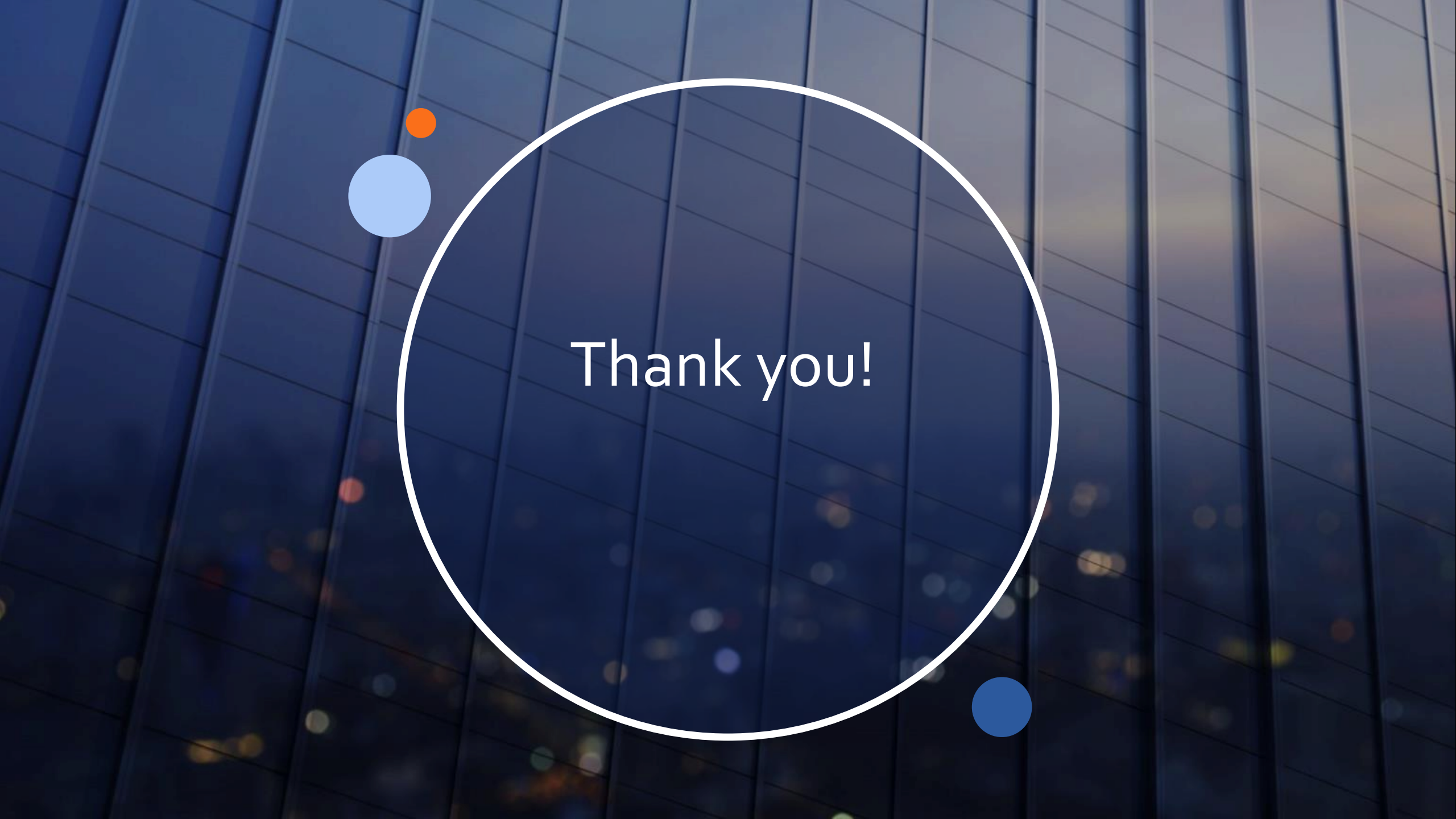


Execution environment used for each quantum programming language



Testing techniques regarding the use, creation, and update of models during the test phase




Thank you!