

# Formalization

Nimit Singhania

July 13, 2022

Formalization is a key process often performed by people when trying to make a decision, when trying to form an agreement between parties or just setting the laws of the land. It is commonly performed by lawyers, engineers, architects, investors etc. It's the process of setting an idea in stone, which is used to make subsequent decisions and resolve disputes. While the word itself conveys a formal process, it is also common in informal settings when families set rules for their children or siblings decide how to share responsibilities among themselves. Such formalization is often necessitated in computer science and software engineering where complex problems need to be tackled. Formalization is a way to concretize the problem and set up the framework under which the solution is found. Often the framework needs to be revised and formalization needs to be updated when a solution can not be found. The steps above are repeated until the formalization is reasonable and a solution can be found.

Formalization is bread and butter for a formal methods engineer. The work of a formal methods engineer often starts with surveying the existing landscape and learning about the underlying system, the core principles and the dynamics (both linear and non-linear) of the system. The next step is understanding the problems and the key problems that need to be addressed. Understanding the problems is essential in order to build a formalization. This is because any formalization will often not suffice and lead to unnecessary complexities not warranted by the situation. The next step is formalizing the system while keeping the problem at hand in the center, and framing the problems in the framework of the formalization. Subsequently, we can go ahead and look for a solution that can work within the formalization. This is useful because it helps breakdown resolution process into two steps: framing the problem under a simple formal framework and figuring out the solution to the original problem.

To understand this better, we shall consider an example. Let's say there is a company where there are two competing parties trying to push their own ideas onto the other with respect to how to tackle a given problem. The first party uses a top-down approach where they would like to work out a solution from scratch helping the company achieve the desired objectives. The other party follows a bottom-up approach that honors the existing practices of the company and would like to augment the existing solution helping us achieve the desired goals while keeping the ownership of the solution. The first party has an upperhand whereby they are in the position to push their ideas, while the

other party has the benefit of being in the company longer and having more faith and trust from the company. The quagmire has reached a standstill with neither party agreeing to the other. Folks are using deceit to push their own agenda while none are coming upfront about it.

There are two issues here. First, the need for professionalism whereby folks must act professionally and not act out violently with others. This is needed to maintain the decorum of the company and present an ethical image to others outside the company (or outside the team). Second, software development is a subtle process that requires calm and stability, in order for folks to be able to think and make progress. It is an intense mental activity which can not be performed when one is at unrest or is agitated. Therefore, it is necessary that we provide folks the space to be able to think clearly and make progress in their endeavours. Thinking from the perspective of the company, it is necessary that both sides come to respect the other and honor the complexity and challenges involved in the process of software development.

Formalization often involves posing the problem in mathematical terms, so that any ambiguity is removed, and the mathematical formalism is a ground truth in itself which can be used to search for a solution. Sometimes, the laws of the land are written in deed, agreement or a contract, while using formal notations that are similar to the mathematical notations used in mathematics. Both these types of formalism try to clear out any ambiguity, ensuring no loopholes can be exploited later by the parties to upend the system or identify a premature solution to the underlying problem. This is necessary because sometimes the parties might be adversarial and might need regulation in order to maintain order and stability within the system.

To conclude, formalization is a process of simplifying the complexities that can arise in disputes, in setting the laws of the land and when tackling difficult problems. It involves setting the underlying terms, often in a mathematical notation, which then serve as the contract between parties and the framework for a solution to the underlying problems. This helps simplify the problem at hand, help ease relationships and make progress. It is a must in formal settings like work, and must be practiced by everyone including software professionals to succeed while maintaining the peace of mind.