

# The Evaluation Scheme

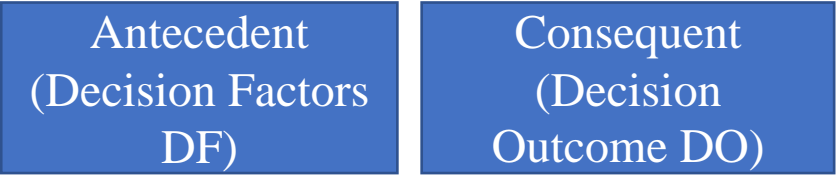
Evaluation labels Definition	Support	Support & refine	N/A	Refine	Contradict
Supports the rule consequent	✓	✓	No data is available to complete the evaluation	✓	--
Supports at least one factor (decision/time)	✓	✓		✗	--
Add a new factor (decision/time)	✗	✓		✓	--
Contradicts the rule consequent or an existing decision factor	✗	✗		✗	✓

# The Evaluation Scheme: Support

- Example (Rule 4)

A startup team should enforce some basic coding standards to be followed within the team

Participant 1	Participant 2	Participant 3
“Yes, don’t make spaghetti code from day one. We have to organize our basic structure, follow the main guidelines, separate the business logic from data layer ...”	“Yes, agree”	“Yes, it totally agree with that. and i believe we are doing the same thing as well. we are not enforcing our developers to 100% follow the standards, but we do have some basic standards ...”

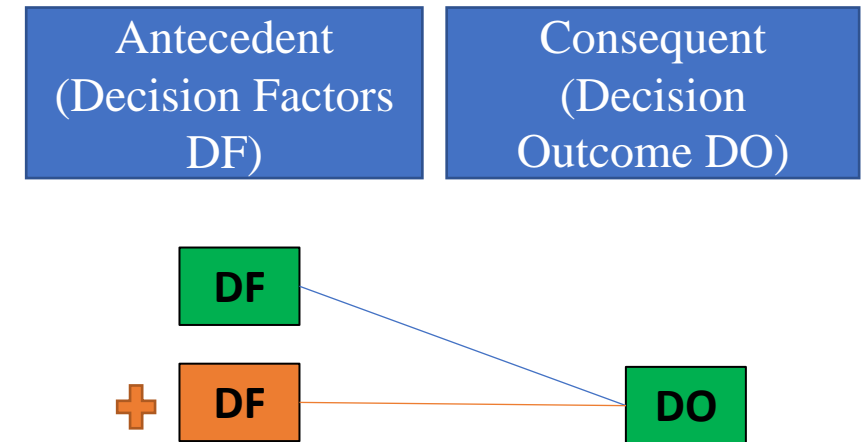


A rule is **supported** if all the participants supported the decision outcome with at least one decision factor that associate with it, but no contradiction or modification

# The Evaluation Scheme: Support & Refine

- Example: Rule 10 (**Before** evaluation)

IF the framework upgrade workload is feasible OR the framework upgrade process is automated, THEN a startup team should upgrade the development framework to the latest version

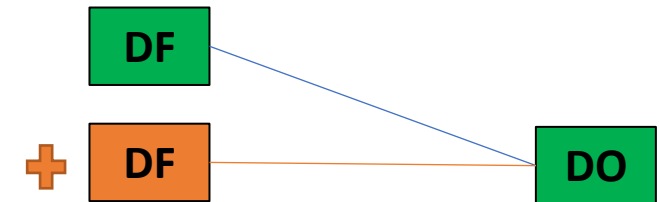


A rule is **supported & refined** if the rule is supported, but any participant has suggested some update to the rule, without contradiction of the original rule.

# The Evaluation Scheme: Support & Refine

- Example: Rule 10 (After evaluation)

IF the probability of changing the framework is low AND the framework upgrade workload is feasible OR the framework upgrade process is automated AND the gap between current and latest version of the framework is high, THEN a startup team should upgrade the development framework to the latest version



Participant 1	Participant 2	Participant 3
if I am a CTO or development manager, I will raise that we have to upgrade whenever a new version come. Ok it is not automated because it needs some code changes to adopt with the new upgrade. So, I'm going to raise it that we need to allocate sometime to upgrade. They might say no we have other important tasks. This will always happen	Usually, we should frequently upgrade. There is no discussion there. In a specific scenario where we are going to stop using the framework fully. So why we go to the hassle of upgrade it if we going to drop it anyway.	In regular basis, yes upgrade the framework to the latest version. We could have limitations where we cannot upgrade the system

A rule is **supported & refined** if the rule is supported, but any participant has suggested some update to the rule, without contradiction of the original rule.