1. Unitization scheme: Factor or Criteria

This scheme is for extracting decision criteria or factors from the interview transcripts. The

* A criterion or factor is something that the participants used as a justification or rationale for a certain kind of behavior. It is used to guide the decision-making process and is a determinate of the choices that are made. The criterion provides direction and resolution to the decision process.
* A criterion or factor is a noun which may have one or more adjectives quantifying it.

**Example**

*“They wanted to be faster to the market, so they took a fork from a pre-made solution.”*

From the example above, the criteria is *“Required time to market”* which is quantified by *“tight”* that requires faster development.

* A criterion or factor is any piece of information that is used when making TD decisions.

In our data, we could not find any factor related to this point. It can be due to the nature of TD decisions that are mostly made intuitively with less usage of explicit information.

* A criterion or factor can indicate certain timeframes in the startup timeline. Timeframes can also refer to certain states or conditions of the product evolution in the startup timeline. We refer to this type of factor as “Time factor”.

**Example**

*“We should keep an eye on the latest version of the framework and do upgrading before the application grows” ... “We could not make upgrade because the codebase becomes huge”*

From the examples above, the criteria refers to a certain timeframe which is before the code size significantly increased (i.e., before high code growth).

**Example**

*“For the integration testing, it is in the plan. But we do not focus on it right now. Or it is not of our top priority right now since there is no traffic on the product.”*

From the example above, the criteria refers to a certain timeframe which is after the usage traffic is high.

* For something to be coded as criteria, it is NOT necessary for the participants to specify exactly how it is used, only that is a driver of the decision in some way.

1. Unitization scheme: Rules

This scheme is for constructing decision rules from the interview transcripts. Note that the unitization of criteria/factors must be done prior to the rule unitization.

* Rules will not appear in the interview transcripts explicitly in the form of IF-THEN statements. They must be deduced from the transcripts in the vicinity of a criterion (i.e., the text references of the criterion). This criterion determines the scope (area) of the text that will be used for extracting the rule. For example, extracting the sentences that contain the factor identified in the previous step.
* Rules guide the decision-making process by identifying appropriate actions used under different set of conditions.
* A rule consists of two parts: the antecedent and the consequent. The antecedent (IF part of the rule) specifies the conditions of the criteria/factors that the participants use as a rationale for their TD decisions. The consequent (THEN part of the rule) determines the appropriate actions that the participants made or recommended, based on certain conditions of the criteria.
* A rule can be associated with one-to-many timeframes. It depends on the timeframes when the consequent (appropriate actions) are made or recommended.
* The use of conjunction (AND/OR). When a rule has multiple criteria or actions, we applied OR conjunction (assuming independence between the criteria/actions used in the rule), except if the data indicates otherwise. If the criteria or actions are related such that they should be use together during the decision making, then we used AND conjunction between them.

**An Example of how a rule is constructed (Rule 13)**

**Case 3 – participant 1**

*“we don’t have that much documentation. Of course, we have some short ReadMe that basically tells how to run the stuck … Now, it is not bad, it is easy to share the information. But I guise it could be pretty near for us when the team is like more than four person”*

Rule 13: version 1

IF the growth of development team is high, then a startup team should improve documentation

**Case 3 – participant 2**

*“the last month or two we started to figure out something with the colleague, we try to make some documentation about how core functionality works. So, if new developers onboard it is easier for them to get started to understand”*

Rule 13: version 2

IF the growth of development team is high (or expected to be high soon), THEN a startup team should document the requirements of some important functions

**Follow-up interviews**

*“I would say like the basic*

*documentation should be at the beginning. At least the setup and local setup procedures. But the very detail and high-level documentation should be at the team growth level, yes.”*

Rule 13: last version

IF the growth of development team is high (or expected to be high soon), THEN a startup team should document the requirements of some important functions (high level documentation)