2.8 Product Verification

The product verification model specifies substantiation (for the whole or a coherent subset of a product model) that:

- product model element instances are mutually consistent,
- the product model is conceptually complete, and
- the resulting product's behavior is a proper realization of the product model.

Product model element instances constitute a network of related content. Consistency must be established between baselined instances having related content, particularly pairs in a specification-realization relationship, and reestablished as instances are modified. Completeness is a determination that a consistent product model is a sufficient basis for realizing an operable version of the product. Verification determines the degree to which a realized product version conforms to its specified product requirements model. This encompasses static verification of product model elements (e.g., reviews), analytic verification of product qualities, and dynamic verification of product behavior

The verification model has five elements[1]: directed reviews, applied analytics, and normative, anomaly, and regression testing. These elements are complementary in determining whether a product realization meets specified customer acceptance and product quality criteria. To this end, the product requirements and product environment models must be consistent with the customer needs specification and all other elements of the product model must be mutually consistent with those. If discrepancies are identified, these are associated with the relevant elements of the product model for resolution.

Directed reviews are an effective means for subjectively evaluating product quality. Analytic methods are an objective basis for predicting relevant quality factors. Testing as a method of dynamic (i.e., empirical) verification is a means for the systematic

[1] Based in part on ***An Acquisition Perspective on Product Evaluation*** (CMU/SEI-2011-TN-007), CMU Software Engineering Institute, Pittsburgh, PA, Oct 2011.

anecdotal evaluation of product value (i.e., utility and quality), profiling dependencies and dynamic aspects of product behavior, and exposing flaws in the product model. Practical and effective evaluation of product value entails the judicious use of all of these elements, as appropriate to its intended purpose.

Directed Reviews

The directed reviews element specifies the results of developer-directed peer and expert reviews in determining the consistency and completeness of the product model[2]. Reviews can focus either on individual element instances or across related instances. A developer initiates review of materials comprising content of a task as needed to evaluate the approach and results per task goals and developmental quality criteria.

Reviewers are selected to provide needed perspectives based on relevant process or subject-matter competence to assist the developer in identifying and resolving uncertainties, inconsistencies, and tradeoffs relative to task and overall product model goals. Reviews should reference the results of other related verification tasks.

Applied Analytics

The applied analytics element specifies the results of applying static and dynamic analyses of product quality, either based on product model specifications or exhibited in the behavior of a realized product version. The product requirements model specifies targeted product quality criteria. The product analytics model guides the use of relevant theory- and heuristic-based predictive analyses of product quality. Analytic evaluations are used to assess the degree to which the product achieves quality factor goals specified in the product design model, not only to improve the design but also to improve the means by which quality factor goals are predicted.

Static evaluations can be applied to alternative versions of a product to evaluate which best approximates specified quality factor goals. Theory-based analyses predict how alternative solution approaches affect specific quality factors and how changes in a

[2] The nature of a "directed review" is described in section 1.3.

given quality factor enhance or constrain other factors. Heuristic-based analyses provide insights based on experience with similar solutions in other products.

Dynamic analyses entail derived metrics based on measurements obtained through monitoring of scenario-based testing or operational use. This generally requires instrumentation of the product or its operational environment to obtain relevant measures of behavioral data that characterize quality factors of interest.

Normative Testing

The normative testing element specifies the correspondence observed between the product model as specified and the behavior exhibited by a corresponding candidate product realization. This correspondence is the result of scenario-based operation of a product realization to obtain a comparison between expected and observed results for a specified testing event. The product model considered may be limited to elements that specify any subset of the envisioned product (i.e., an assembly of related components) sufficient to perform a specified scenario.

The product requirements model defines the behavior that a realized product is expected to exhibit. Evaluation criteria for a test scenario considers how well it addresses product behavior as specified in the product requirements model.

Testing Platform

A candidate product realization is deployed into an evaluation platform to create a testing environment that is an approximation of the product's actual ecosystem (as specified in the product environment model) and computational platform. Capabilities for instrumented monitoring and control of testing events may be built into the product as realized or may operate by means of evaluation platform mechanisms. Instrumentation includes the means to inject data and collect, retain, and report otherwise unobservable computational results.

{creating a testing infrastructure based on oper env defn; instrumentation mechanics; data collection-reporting-auditing; logging/data capture; family of testing platforms, scenarios, or events}

Testing Events

A testing event occurs on an evaluation platform with an operable instrumented product realization of any component assembly. A component assembly is a subset of a baselined product instance, for verifying one or more components along with any other supporting components needed to operate on the evaluation platform.

The environment and any associated entities may be virtualized and any devices that are elements of the product may be emulated if they are not physically accessible. Both the product and any simulated or emulated elements may be instrumented to artificially control their operation and provide observability for tracking the causes and effects of their behavior.

Testing has three stages: preparation, execution, and analyses of results. Preparation is the plan and setup for a testing event. This entails an objective, customer-based or freeform scenario, a guide/playbook to perform the scenario, an evaluation platform set up with data initialization scripts, and expected results. Execution is the recording of a manual and/or automated performance of a planned testing event (with automated, directed, or device/user-role portrayal), including results of monitoring and control actions and collected data on behavior. Analyses of results correlates testing event occurrence to planned, noting any divergences from expected with suspected cause and any remediative or diagnostic actions for either the test event or its subject. Reporting should include results of any analytic evaluations of the testing event or resulting data.

Scenario Specifications

A primary basis for testing events is the set of operational scenarios specified adjunct to the customer needs element of the product delivery model. A scenario is specified in a form that provides sufficient information (preconditions, actions, and expected effects) to define a testing event. A scenario must specify conditions and data for platform initialization and correct observable behavior that should result. Results include any interactions that are expected to occur between the product and any entities in its operational environment and consequent effects on associated information. Scenarios should be conceived both to confirm occurrence of expected behavior and detect

occurrence of unexpected behavior (e.g., due to invalid data values or as unforeseen effects of proper behavior).

Anomaly Testing

The anomaly testing element specifies results of testing based on scenarios that express degraded, abnormal, or unexpected behavior of the product. An anomaly test is a dynamic evaluation as to whether (1) the product behaves correctly under conditions of failure (e.g., inaccessibility of an entity or platform service) or fault (i.e., of the product to operate correctly in response to failure), and (2) the product does not accommodate behavior that is either unspecified or expressly excluded in the product requirements model. Anomaly testing supports root cause analyses for determining the source of observed flaws in product behavior.

Regression Testing

The regression testing element specifies results of testing to ensure that nominally unchanged aspects of behavior have not been affected by changes in other aspects of behavior. Regression testing is repetition of prior normative and anomaly testing on product capabilities that have not been changed and therefore should exhibit identical effects as produced by the prior version of the product. A regression test that produces different results indicates that the product has been modified relative to the specified test. This may indicate that other changes have had unexpected effects on unchanged parts of the product. Alternatively, the test specification may need to be changed to better account for expected effects of legitimate changes.