2.6 Product Analytics

The product analytics model specifies a collection of analytic models appropriate to the problem-solution space of an envisioned problem. Such models may be quantitative or qualitative, objective or subjective, based on measurements or anecdotally derived.

This model provides the means to predict and refine properties of a product based on product model content. Generally, each specified analytic model will provide predictions related to specific product quality factors that are identified in the product requirements model as being of significance for the product. The information needed and the accuracy of resulting predictions depends on each of the particular models used.

This model is a medium for evaluating the suitability of a product instance relative to the product requirements model, for understanding how changes in the product model could affect product quality factors, and for comparing product model alternatives that resolve uncertainties and tradeoffs differently so as to inform what solution is the best fit as a whole to the problem defined by customer needs.

{what quality tradeoffs are being made? what are the effects/side-effects of this? what emergent behaviors are likely and what changes would change those?}

- *quantitative & qualitative models of system/product behavior*
- *sensitivity-based projections of behavioral qualities*
- *rqmts/design/build alternatives and tradeoff analyses*
- *compare multiple alternative product builds for best-fit to needs*

The expected behavior of software is not only a question of exhibited functionality but also how all relevant operational qualities will be satisfied. This entails making tradeoffs among qualities, for example, achieving safety and security to an acceptable degree may take priority over some aspects of functionality or usability. There currently exist only limited means to predict, and in some cases even objectively measure, the qualities that software will exhibit in operation. However, both what is possible now and what greater capabilities will emerge over time are worth exploiting to avoid having to make changes after deployment to fix quality defects that could have been discovered during development. The goal of the analytics model is to establish a common basis for defining the quality of software as it is being built, influencing how it is built, and setting customer expectations concerning what is known and what is uncertain concerning quality factors of software in use.

The properties that a software product is expected to exhibit are defined in the product requirements. How these are to be met and the effects of both enabling and inhibiting dependencies among them are defined in the product design. The purpose of the analytics model is to be evaluate the degree to which the product as it is being built is able to satisfy the envisioned quality factors, both in theory and relative to a specific operational environment.

The least sound but least costly means of deriving analytics are the insights gained in verification of product elements through reviews and testing. The value of these insights will depend on the competence applied in producing the verification model. However, developing analytics content based on analyses of the verification model are beneficial in evaluating and improving the degree to which software is likely to meet quality expectations.

Product Quality Metrics

Product requirements specify and prioritize product (i.e., behavioral) quality objectives[1]. Product analytics start with these objectives to more precisely characterize the quality space in which the product is to reside. This entails determining the relative importance of identified qualities and the acceptable limits that each quality needs to meet. Alternative resolutions of the resulting quality space are analyzed in terms of how various qualities interact and the cost to realize each combination.

---

[1] Conversely, process (i.e., developmental) quality objectives and their evaluation are defined in the project management model.

Each individually identified quality defines a facet of the quality space, corresponding to some distinguished property of the envisioned product. Within the overall quality space, each facet can be defined and measured independently but changes in the value of any facet can cause the values of other facets to change as well. Understanding the relationships among quality facets is a key aspect of product analytics, providing the basis for adjusting the quality of the product as a whole.

Product analytics specifies the degree to which each quality factor identified in product requirements is to be addressed by the product. Failure to satisfy even less significant factors may result in a product that fails to meet customer expectations. Similarly, a failure to include such factors in product requirements should be identified as a discrepancy to be resolved.

As an example of how qualities can interact, a product may be specified as needing high usability, moderate performance, and moderately high security. Each of these are then defined here more precisely in terms of the range of values associated with each and how they are to be measured (quantitatively and/or qualitatively) and how these qualities interact, in particular how changes in one quality affect the others. In this example, these security and usability expectations may both modestly reduce performance. If usability and security goals are projected to be satisfied in the resulting product but performance is projected to be unacceptable, either some aspect of the usability or security goal that impact performance might be possible to relax or alternatives for increasing how performance is achieved would need to be explored.

Product Instrumentation

Analyses of how well software meets product quality goals requires a means to gather data corresponding to the behavior of the operational product. Some data about a product is associated will the externally observable behavior of the product and can be collected by software operating external to the operational product itself.

Other data can be collected only by modifying the product itself to include instrumentation that enables the monitoring of its internal behavior. Such instrumentation of a product could be specified as a required part of the operational behavior of the product but is often only included as part of a distinct developmentally useful version of the software.

The detriment of such developmentally useful instrumentation (and to a more limited degree, external monitoring) is that it can reduce the degree to which certain qualities (e.g., performance or security) will be satisfied in the operational product.

Formal Models

(analytic predictive-metric goals and models)

Defects and Root Cause Analyses

{observability: trace/diagnose/understand source/cause of defects} {hard: product is doing what it should; harder: product is capable of doing only what it should/is expected, particularly for reused components}

Alternative Products Evaluation

{multiple product versions: best-fit to needs, eng tradeoffs}