2.3 Product Requirements

The product requirements model specifies the behavior that a product is expected to exhibit, within a specified operational context[1]. This specification of product behavior is based on the product delivery model specification of customer needs and the product environment model specification of external entities with which the product interacts.

During development, the requirements model is an evolving "build-to" specification of the product, defining all aspects of the observable behavior that the product must be built to exhibit. Upon deployment of the product into operational use, the requirements model declares the correct "as-built" behavior of the product. The evolution of a product is viewed in terms of the aspects of its observable behavior that are changing.

Whereas the customer needs element of the product delivery model describes the capabilities the customer expects the product to support, the product requirements model prescribes the product's expected observable behavior from the developer's perspective; this has 5 elements:

- Concept – a concise description of the product's purpose, capabilities, and utility
- Context – a specification of the ecosystem in which the product will operate
- Content – a specification of the behavior that the product will exhibit
- Quality – a specification of the criteria against which the product is evaluated
- Constraints – a characterization of extrinsic factors that limit the product's realization

Requirements Model: Concept

The concept element is a concise summarization of the capabilities that the product is expected to provide. It concisely defines the rationale for a product, based on customer needs as specified in the product delivery model. Its purpose is to establish a shared developer vision of the product's intended purpose in enabling the customer to achieve their objectives. The product concept should suffice in describing any realizable product

[1] Loosely derivative of: K.L.Heninger, "Specifying Software Requirements for Complex Systems: New Techniques and Their Application", IEEE Trans on Sw Eng SE-6(1), Jan 1980.

that conforms to customer current and likely future needs. This should be augmented with definitions of any specialized terminology and references that support understanding of the problem and its solution.

Requirements Model: Context

The context element is a specification of the assumptions made concerning the nature of the customer's operational environment. This identifies the entities with which the product will interact to share information and coordinate actions. Details concerning potentially relevant external entities, including devices, system interfaces, and user roles, and their properties are defined in the product environment model. The environment model specifies the environment as a dynamically changing information space and entities as the means by which it is sensed and influenced. Each entity is specified there as a logical device according to its data and services responsibilities relative to the environment or its own operations. An entity may also selectively provide indirect access to data and services of other entities. The context model specifies the product's dependencies on each relevant entity, to be accessed via its computational platform.

Requirements Model: Content

The content element is a specification of the behavior that the product is expected to exhibit, relative to its specified context. Behavior is expressed in terms of the effects that the product initiates by means of associated device capabilities. Each effect is a function defining the value of an output as a computation over an information model that represents the product's knowledge of its computational circumstances.

{formally define "content" incl modes/user-roles, invocation-/event-/demand-driven behavior, dictionary of terms, ?}

Requirements Model: Quality

The quality element is a specification of a the product quality factors that any product realization must satisfy to be acceptable. The relative importance and sensitivity of the various factors provide a basis for tradeoffs among alternative solutions. The product

design will determine the degree to which the significance of the various factors will differ in the product realization.

Requirements Model: Constraints

The constraints element is a specification of provider-established, customer-determined, or externally-imposed contextual factors that limit options for the product's realization. Potential constraints include: aspects of the operational (deployment) platform, enterprise policies and practices, regulations:legal/contractual, and customer conventions or preferences. Extrinsic constraints (i.e., not inherent to a product's acceptability) that cannot be relaxed may preclude otherwise potentially preferable alternative solutions.

<----------

To ensure a proper understanding of customer needs, the development project needs access to persons knowledgeable in those needs and the ways that customer enterprises work. Developers themselves may have significant competence in the area. There may be people who are part of program marketing or who are representatives designated by one or more customer enterprises as having competence and authority to speak for the enterprise. Customer-designated representatives may be part of a formally constituted product acquisition effort. The requirements effort will confer with these customer representatives for help in resolving uncertainties concerning context or needs and in providing feedback on how these are to be realized in a product.

To ensure a focus on a proper understanding of customer needs and how well the product will satisfy them, some persons should be designated as customer representatives. Such representatives, viewed loosely as "the customer" and designated as having competence and authority to speak for customer interests, may come from one or more actual customer organizations, customer-designated acquisition organizations, or the developing enterprise's marketing or sales organization. Any development activity may need to confer with the customer concerning the context in which the product will be employed.

Building the right product requires an understanding of the customer enterprise, how that enterprise operates, how the envisioned product will affect that operation, and relevant subject matter. Such understanding comes most easily from having worked in such an enterprise or having built products that solved similar problems, recognizing how differences in customers' circumstances can affect a solution. This understanding may be supplemented through customer-provided information, or from program marketing as a customer proxy. Needs that are initially incomplete, not well understood, or poorly communicated will require refinement through customer-provider collaboration as the product is developed.

From a developer perspective, customer needs is an expression of how the customer expects the product to support their activities. This is only an approximation of their actual needs, incomplete and possibly flawed, and an "under-specification" of the product in that many different products could be built that would satisfy those needs. Without exploring all the conceivable products that might satisfy a customer's expressed needs, the objective of development is to realize a specific product that reasonably fits actual customer needs (possibly changing as development proceeds), resolving any tacit assumptions, omissions, ambiguities, or uncertainties in the expression of those needs. The resulting requirements model is an "over-specification" of the product that is ultimately deployed into operational use, with the intent of eliminating developer uncertainties as to what is to be built by providing a shared understanding of expected product behavior.

---------->

As a product is revised in keeping with changes in actual (or previously understood) needs and technology, the requirements model expresses how those changes affect product behavior or operational context. The requirements model can change at developer discretion, as long as it ultimately achieves consistency with expressed customer needs:

- As development progresses incrementally from a partial to a complete solution
- In recognition of improved developer understanding of actual customer needs

- As a result of feedback from work on dependent model facets

The requirements model can change as development progresses as a result of negotiation between project management and the customer due to:

- Customer-initiated revisions in specified customer needs to better reflect actual (possibly misrepresented or changed) needs
- Management-initiated changes in previously agreed customer needs, in response to project-identified impediments (e.g., feasibility, cost, or delay) to product deployment

Information on Which Requirements is Based

The product requirements model is conceived based on two primary sources of information:

- Business area competence – the *knowledge* of the developer in the theory and practice of customer and similar enterprises; familiarity with *experiences* with similar and related systems and in developing similar and related products; *expertise* that the developer has with relevant techniques and technology appropriate for developing software of the sort needed
- Customer needs – a customer perspective (conveyed by the customer needs element of the product delivery model) on how their needs are perceived in terms of the purposes that a product will support, the capabilities that the product will provide in support of their operations, and any constraints that an acceptable solution must satisfy

Customer needs, being an expression of customer criteria for product acceptance, is an under-constrained specification of the envisioned product in that many different products could be realized that would satisfy such criteria. To the degree that customer needs under-specify a product, the developer has flexibility to make engineering tradeoffs for a viable best-fit solution. By applying business area competence to customer needs, the product requirements specifies the observable behavior of a specific product to be realized. The result is an over-constrained specification of the envisioned

product, reflecting developer judgement in identifying and resolving uncertainties, alternatives, and tradeoffs while elaborating and refining developer understanding of customer needs. This has the effect of reducing a candidate set of potentially acceptable products to a single product to be realized. In doing this, the developer may suggest changes in customer needs that will make better or less costly solutions feasible, leading in appropriate cases to a developer-customer collaborative revision of the customer needs element.

<————————

Customer An expression of customer needs may not include all the information that developers need to build a suitable product. expression of “needs” is typically incomplete, possibly flawed versus actual needs, and overly constrained in how new practices or technology might enable an improved solution. refine understanding of customer needs, through a collaborative dialogue between developer and customer, to clarify what capabilities are actually needed and any constraints on product realization.

Customer needs define the criteria for whether a resulting product provides envisioned capabilities while satisfying operational constraints. While customer needs can arbitrarily constrain any aspect of a product’s realization, unnecessary constraints can increase the difficult of development, and consequently the time and cost for completing the product. An ideal expression of needs avoids arbitrary constraints on a potential solution, leaving the developer flexibility to consider alternatives and tradeoffs that will lead to a viable solution.

The requirements model definitively focuses the rest of the development effort by specifying the exact behavior of a product instance that is being or has been built.

The purpose of explicitly defining customer needs is to establish a shared understanding between customer and developer concerning the envisioned product’s capabilities. gives developers the flexibility to explore alternatives, resolve uncertainties, and make tradeoffs.

An important aspect of the product development effort is to expose errors and uncertainties in the expressed customer needs in order to allow the customer to refine

their expression of needs that the product will be expected to satisfy. The means for this is to ensure that the requirements specification is consistent and complete with respect to the customer needs as expressed. When development activity exposes possible issues in the expressed needs, the requirements action is to confer with customer representatives on how to realign needs and requirements, by revising expressed customer needs and/or modifying the requirements specification.

Customer needs are effectively an idealization of key aspects of the product being built, in the sense that many different products could satisfy those needs. The purpose of the requirements is to describe the product that is actually being built. Furthermore, a product is developed in a series of increments, which will much of the time during its development lack specific capabilities that the product release will have. During development, product requirements may differ from expressed customer needs for several reasons:

- The requirements may differ from described needs that are pending revision due to discovered differences from actual needs
- Some aspects of needs may have been deferred as being outside the scope of the current development, to be addressed in a future increment or product version
- The complexity of some aspects of needs may not currently be viable to address, requiring additional knowledge, expertise, or technical means, exceeding time and cost goals, or possibly warranting a negotiated revision of perceived needs

When the requirements model response to needs results in unforeseen complexity, portending added effort, the developer has the option of proposing revisions in customer needs to mitigate that complexity. This allows the customer to influence the tradeoff between capability and effort. Alternatively, the developer may create an interim more limited version of the product, that can later be revised to more properly address the need.

Another reality of aligning requirements with needs is to establish a distinction between expressed needs and the extent of those needs that an interim version of a product is able to meet. A sound development effort will first build a product version that

minimally meets a subset of expressed needs, with subsequent efforts leading to more capable versions.

————————->

Product Behavior

The product behavior specification defines how the product produces observable behavior. This specification is organized in terms of the product's actions upon each of the logical entities to which it has access as specified in the product environment model. These entities are categorized as edge device, user role (engaged via associated interface devices), or system (viewed as an aggregation of associated logical entities).

Behavior is expressed in a formalism that is characteristic of the sorts of interactions that a given type of entity accommodates (i.e., the actions that it recognizes as defined in the product environment model). Each such formalism has two elements: the criteria by which the action is initiated and how the content of the action is determined.

The product environment model specifies the entities that are accessible to the product and the set of addressable actions associated with each entity. An action is the dispatch of a request and associated data. Each action is associated with a particular type of device (edge, user role, or system) whose specification determines the form that valid actions can take.

{This will be described more clearly once the content of the product environment model is completed. Defining behavior this way is easy to do but not so easy to explain without getting overly specific to particular formalisms.}

Action Criteria

Action criteria is a specification of the operational condition that causes the action to be initiated. Actions can be initiated based on one or any combination of (1) a specific request (i.e., event) initiated by another action, (2) a transition in the value of a predicate concerning the content of monitored data, or (3) a transition in operational mode of the product instance. When given criteria is satisfied, the associated action is initiated according to its specified effects as appropriate to the type of device it targets.

Edge Device Actions

An edge device initiates actions that may affect the environment as needed to realize intended goals of the product's operation.

User Role Device Actions

A user role device presents data to convey information within the purview and operational context of users performing the activities associated with a designated role. A user role device is realized by means of one or more user interface devices. Data may have associated mechanisms associated with a given type of user interface device for responding to user actions.

System Device Actions

A system device transfers action requests or data to one or more specified interface devices defined as associated with a specified type of system entity. Each interface device is defined in terms of the form in which understood actions and data can be accepted.

Product Quality

The product quality specification defines the criteria by which the product is built and evaluated as acceptable for its intended use. Product quality is the degree to which a product provides the capabilities that are efficient and effective toward meeting the needs of a targeted customer. The quality of a product is the aggregate of the quality of its elements. Product quality has several facets, some representing the form these capabilities take but others concerning aspects that enhance or limit those capabilities. These facets are organized into four broad categories for the quality factors that mutually characterize a product. Each of these categories encompass a flexible set of finer-grained properties that each to varying degrees influence the acceptability/fit of a product for its intended purpose. While all four categories affect every product, the specific encompassed properties and the degree of influence that each has on acceptability of the product can differ based on the nature of the product. The requirements model defines which specific properties apply to that product and how

much flexibility there is in each property to enable tradeoffs among properties to resolve conflicts among them.

{explain how quality factor goals are expressed as objective guides design-implementation analysis of alternative tradeoffs and product evaluation, either as a "lower-bound" or value range.}

{explain how quality factor goals may differ across different parts of a product (e.g., a factor such as safety may be addressed by only a portion of the architecture)}

{note that "functionality" is a behavioral quality factor in that it is may need to be traded against other factors (i.e., functionality restricted to achieve e.g. security, safety, or performance goals).}

Notional Behavioral Qualities (1)

Functionality, the degree to which a product provides expected capabilities to the customer (i.e., effectiveness); typical elements include:

- *Utility* (satisfies its intended purpose)
- *Interoperability* (operates and communicates properly with external elements of the environment and ecosystem)
- *Adaptivity* (has the means to modify its own behavior as directed (i.e., reconfigurability) or in response to prescribed circumstances)
- *Reproducibility* (exhibits differences in behavior or information content only as a result of differences in external influences)

Performance, the degree to which a product supports an anticipated workload, consistent with available resource capacities (i.e., efficiency); typical elements include:

- *Responsiveness* (reacts to external stimuli and internal effects sufficiently to maintain consistency internally and with respect to its ecosystem)
- *Utilization* (makes best use of available resources, including energy and environmental efficiencies achievable within equipment capabilities)
- *Conservance*: minimizes use of physical assets and logistical resources and impact on environmental resources and conditions
- *Throughput* (produces effects at a rate sufficient to maintain consistency with external elements of the environment and ecosystem)
- *Scalability* (adjusts to projected variations in workload and resource capacities)

Notional Behavioral Qualities (2)

Dependability, the degree to which a product continues to produce expected effects (behavior and data) under all (normal, abnormal, and unforeseen) conditions; typical elements include:

- *Reliability* (exhibits consistent and accurate effects under normal and degraded conditions) {includes predictability, stability, and survivability, including resilience and recoverability}
- *Availability* (operates without interruption or unavailability of information) {includes continuity and connectivity}
- *Integrity* (provides acceptably accurate and complete information, while excluding unauthorized access) {includes fidelity and privacy/security, including information assurance and cybersecurity}
- *Safety* (prevents or detects and mitigates conditions or actions that would cause unintended damage to any part of itself or its ecosystem) {includes avoidance of misrepresented, distorted, unethical, or biased content}

Usability, the degree to which a product is able to be used properly by its intended users/operators; typical elements include:

- *Conformability* (operates consistent with the competencies (language-terminology, knowledge-expertise, and skills-abilities) of users) {includes accessibility}
- *Learnability* (aids and instructs users in the nature and use of its capabilities and information content)
- *Explainability* (is able to communicate causes and rationale for prior or prospective behavior and information content)
- *Aesthetics* (presents and properly conveys provided information and functionality in a form that facilitates intended usage)