5.1 Basic Technical Challenges

The producibility vision is achievable with the current abilities of competent developers. Techniques and supporting technology have gone through significant improvements over the years. Further advances in many aspects of software development practice would enable further improvements in the quality and effectiveness of software-based products. Improvements in support of challenging aspects of operational infrastructure would enhance productivity and quality for needed products.

Important opportunities for improvement fall into two categories: developmental practices and enhanced operational capabilities.

{correlate to Producibility paper research topics?}

Developmental Practices

Opportunities for improved developmental practices concern improvements in practices that developers use in building a software-based product or product family:

- Collectively revising and redeploying a set of similar deployed products, for both feature enhancement and defect diagnosis-repair, adjusting for differences resulting from each product's defining application model as well as customer-specific configuration settings and user-specific preference settings.
- Verifying that product modifications, whether due to product-specific specification changes or product family-level modifications, do not degrade whole-product consistency and completeness.
- Enhance tools that are elements of the development platform to properly support simultaneously building multi-version products and multiple similar products for evaluating alternative solutions to the same or similar problems, with associated rationale for differences and their observed implications on each product's behavior.

- Provide capabilities for determining both retrospective (as-built, descriptive/analytic) quality and prospective (to-build, prescriptive/predictive) quality.
- Formulate essential tacit and heuristic (domain or circumstance dependent) knowledge explicit (implying managed uncertainty, incompleteness, and error) with rationale, allowing it to be questioned and progressively refined (analogous to viewing commonality and variability assumptions as ground truth for a family).
- In modeling ecosystem reality, apply methods to ensure data integrity, currency, and consistency: account for the degree of uncertainty due to inaccessible/unavailable, missing/delayed, invalid, inaccurate, or imprecise measurements, enhance using corrective approximation of actual value; correct for value divergence due to insufficiently current (aged) measures (e.g., associating reduced confidence or interpolating from past values); maintain dependencies among values to avoid or correct inconsistent values (revaluation to replace dated or inconsistent values, using higher confidence or multiple information sources to infer better value). {info/data confidence, uncertainty/unknowns (costly to determine, unknowable, or falsified or suppressed due to secrecy/privacy/competitive risk)}
- Mitigate conceptual, logical, and physical data representational impermanence (extensibility with computational efficiency given unforeseen need for info, subject content or metadata); (tradeoffs: storage/access efficiency, representational diversity, evolving schema for data and metadata, efficient for data values/relations/groupings)
- Accommodate terminological flexibility (language constraints, different terms/same meaning, same terms/different meaning - definitional context) (specifically, tailoring terminology to fit differing customer or market standards or conventions)

- The ability to characterize a product family instance in terms of both common and customer choices regarding changing needs, operational context, and evolving domain capabilities
- The ability to trace engineering tradeoffs to customer-determined quality criteria
- The means to determine how decision resolutions affect product quality factors
- Methods and tools for verifying a product family in aggregate as well as with built or example instances
- Practices for adjusting to an unbuildable product specification, identifying and choosing among approximate fit alternatives in a product family (i.e., finding closest matches to buildable products by minimal restrictions on customer decisions)

Operational Capabilities

Opportunities for improvements in operational capabilities concern enabling developers to build products with these capabilities:

- Detecting, mitigating, or managing hardware failures (delayed or infeasible recovery, data lost or delayed, behavior constrained)
- Observability of behavior supporting determination and management of operational constraints due to faults and failures; recognizing atypical but predicted/correct versus unexpected behavior
- Operating with degraded behavior, to account for potential hardware failure, software flaws, erroneous or missing data (constrained but planned and supported versus unpredictable effects); managing resource-constrained behavior, suspending or deferring non-critical behavior
- Introspection-based explanation and rationale for behavior (with operator dialog, logging, or trail of cause-effect evidence); self-monitoring of behavior to detect or preclude any unspecified (out-of-scope) behavior (against what definition? related to explanation?)

- Hardware-independence for late selection of hardware appropriate to specified behavior (cost, security, accessibility, locality, latency, capacity, …); employ local, peer-to-peer, and multiple layers of server processing capability (every device is an encapsulated data service; some affect the environment, directly or indirectly/via human/hardware: information being where it is needed when needed/with acceptable delay)
- Temporal implications for behavior: local/universal time, transience/persistence (time is local, strategy/tasking can be across time but tactics/actions must be local; even "real-time" must account for latency; allowance for inherent delay/latency/variance in need for zero-delay or otherwise unsupportable "instantaneous" communications)
- Mitigating information/communications overload: data filtering & promotion (what data is relevant/needed & sent/stored to provide capabilities at progressive levels of processing) {for immediate need versus for historic/retrospective/learning} {what to do with input from millions of petabyte enviro-sensor data streams: at each level, use? then discard or store/forward? reference implications of time/latency}; filter data for contextual relevance to mitigate for information overload (current/historical/projected values: currently ad hoc problem-specific choice)