Context Variation by design (CVD): product development for the 21st century

CVD (Context Variation by Design) is a product development approach that has as a starting point that the complexity of contemporary society should be acknowledged and worked with to achieve better results. More often than not, a design challenge is (over)simplified to make it ‘manageable’. In particular the focus is often on one use-context in order to respect the micro specifics of that context. This is tempting and seems sensible. However by severing the ties of the focused scope with the rest of reality, such an approach is increasingly likely to result in solutions that have limited relevance and that cannot easily be scaled to new contexts like regions, countries, segments. CVD purposefully makes use of the reality of complexity instead of simplifying it away. The approach and associated mind-set includes a number of guidelines but is not intended to provide a step-by-step method or instructions. Many existing tools and design methods are available to designers to use and choose from, fuelled by the designer’s own creativity. Prescribing the CVD approach by setting out exact steps would take away the designer’s autonomy and creative freedom.

The key characteristics and associated benefits of CVD are:

- **Accept reality and make it work for you:** By being open about the reality that problems are not isolated to a singular context, the multiformity of the problem is respected. You get more diverse insights by connecting subjects (i.e., people and their insights) from the different parts of reality. In this way you build collective intelligence, instead of (just) very specific mono-contextual intelligence. After letting these insights interact in a ‘shared solution space’ you can still decide how to deal with conflicting or ambiguous sets of requirements. One can still decide on context-specific adaptations, but then born from an intentional shared solution space instead of born from one solution that was developed for and happened to be successful in one context.

- **The right problem formulation:** Having a mind-set that respects the reality of complexity enables you to recognise its connections within the larger societal system e.g., “How can needs for electricity in areas with technically unreliable power supply be addressed in a way that is affordable for end-users and financially attractive for a company” as opposed to “How should we market product X in context Y to be profitable in Z years”. The latter version considers society as an unambiguous, closed box. What some might perceive as strength (focus) is a crucial weakness: no eye for connections with broader reality and thus possibility of keys to the solution residing there.

- **Rich design space, not universal solutions.** The so-called shared solution space is not aimed at developing a ‘universal solution’. Instead, a set of core insights on the problem, products/services, segments, marketing and business model, partners etc. is achieved that benefits from the richness of the variety of inputs and interaction between these inputs. The result of this process can then be translated into actual solution concepts for one or more contexts. The eventual contextual variations of the solutions can be reflected in different parts of the overall solution, which can for example be modular (in/out) or flexible (adjustable): functional features, overall design, but also communication, marketing, pricing, distribution etc. There are also likely connections and synergies between these variations that would not have been achieved without a simultaneous exploration. An obvious example is an intentional multi-context business model (e.g., revenues from version in one context cross-subsidise sales of same product in other context).

---

1. This includes services and product-service systems
• **The main benefits**: 1) a rich solution space, 2) a superior base solution and 3) a cost-effective expansion (i.e. scaling or adaptation) to more contexts with *informed* contextual variations, working from that base solution space.

**CVD-design principles**
The CVD-approach uses four design principles that also interact with each other. Together they create a design dynamic that provides some guidance. The third principle is especially useful at the start of the convergence phase.

- Decomposition: address a complex design challenge by accepting the multiformity and distinguishing aspect systems instead of sub-systems. Breakthroughs for solutions will lie in the connections between different elements rather than in separating them.
- Systematic Variation: vary the elements of the problem and solution directions along the dimensions of product, market and network. Varying the actual decomposition is one, daring, example of such variation.
- Satisficing: acknowledge the limitations of “optimising” and undesirability of “compromising”, focus on *satisficing* to create the starting point to build from to achieve the best possible result in the face of a variety of requirements.
- Discursiveness: acknowledge the multiformity of the situation in time, by allowing jumps and turns in the process instead of only steps and iterations.

**A Management approach to match**
To be able to work in the way as described, the (project) management approach needs to be aligned. In practice this means that especially in the first stage, when there are many uncertainties, project arrangements need to be such that:

- Respect that complexity implies unpredictability; design the process so intent is used as guiding factor so uncertainties do not cause negative distraction.
- It is realised that people make or break the eventual success (i.e., successful innovation requires much more than a product push).
- It is useless to set ‘smart’ targets if you have no knowledge to base these on.
- Dialogues about identifying the best options should be generative instead of transitional (convincing others of one reality).

**The main concerns addressed**
Based on reflection of a diversity of design researchers and practitioners on CVD, a few recurring concerns have surfaced. We address the main ones here:

- **Q**: Is CVD an attitude or a method? For an attitude it is ‘too logical’, for a method it is not concrete enough. **A**: CVD is best considered as an approach and general mind-set on how to tackle complex design challenges. It seems to make some intuitive notions more explicit. It has proven to give direction to designers who use it. With more experience, more practical guidance can be provided. Detailed instructions kill creativity however.
- **Q**: How do you know which contexts to include, when to stop diverging, how to let insights interact and when to start converging? **A**: these are valid questions. Experience shows that practical aspects like company scope, time allowed and openness of the team highly influence the answers per case. Including 2 or 3 contexts instead of just 1 is already a step that adds much value.
- **Q**: will the extra time needed at the start be feasible in practice? **Managers, investors and small companies won’t like this use of extra resources, nor the**

---

2 First discussed by H.A. Simon, *The Sciences of the Artificial*
risk of uncertainty. A: if you are aiming at developing a solution that can be scaled, adapted or replicated across contexts, eventual costs and risks of a sequential process will be even higher. Acceptance of a higher use of resources at the start depends on the ability to see the bigger picture. CVD appears to have most added value for developing new concepts, somewhat less for simple line extensions.

Use of CVD in practice: two examples
What would working via the CVD mind-set and method result in? Below two examples are described based on practical experiences in using CVD: Exact company or product details have been omitted because of confidentiality.

Example 1: an intentional multi-context business model
The problem concerns a very specific target group in a developing country. Ordinary product development resulted in a product that could work ‘technically’ but would never be affordable for this group. Working with a CVD, a different approach was used: by taking the eventual business case into account at the start (discursiveness), the designers saw that the same problem in fact is also experienced by other, more affluent, target groups (other angle for decomposition). These were then included in the exploration. Some requirements turned out to vary, but some were quite similar. Furthermore by “confronting insights” a much richer solution space was created. For example in one affluent context there was experience with a solution-direction which could also be used in the initial low income context where the problem was going to be tackled. Vice versa, the specific user-interaction aspects were stricter in this initial context and addressing these needs created a better idea for the more affluent context. This “insight interaction” would not occur when “transferring learnings” from initial to secondary context
In the end, the main concept that was developed entailed a prototype for the initial context, which was enriched by information from the other contexts; in terms of the business model, these contexts were included in the roadmap to increase the financial viability: the scalability was immediately an important factor. Another “shared aspect” is the story behind the product which results in a (diversified) marketing narrative, while the product can remain largely similar (satisficing) and the steps to be taken to adapt the product for the affluent contexts are roughly known.

Example 2: segment distinction creates a modular wide range solution
A Western company is interested in a promising foreign market. However, instead of exporting their own product or developing one product for the entire market, they realise that it actually consists of many potential segments (decomposition). Based on first explorations they identify 6 different segments (i.e., contexts) that seem most promising and relevant to include in the exploration phase; after exploring a variety of aspects of these contexts with the use of various partners (systematic variation), and analysing which combination of contexts can still be served in a sensible way, they decide to focus on 3 contexts. These 3 represent the most extreme cases that still demonstrate some overlap in requirements on a number of criteria (satisficing). Furthermore, as a non-coincidental upside the company has distribution candidates for these contexts within reach (discursiveness).
The eventual solution contains a product with several modules, with different combinations for each context. The different insights from the shared solution space have resulted in design implications that can be called “generic”, flexible (extent of inclusion) or variable (modules). In other words, some shared insights contribute to generic parts in the solutions, other insights to flexible or variable parts.