Development Models: Blending Agile Practice with Concurrent Engineering in Physical Product Development

Review paper for Total Life Cycle Management, Mechanical Engineering · Domain 1: Development models

Keywords: integrated product development; Total Design; concurrent engineering; set-based concurrent engineering; agile product development; Scrum; SMEs

1. Development Models

Every product that reaches a customer, whether an e-bike drive or a robot-mounted riveting tool, is the result of a long chain of decisions about what to make, how to make it and when to change course. Pugh's Total Design treats that chain as a single integrated activity running from market need and specification through concept and detail design to manufacture and sale [3]. Integrated product development carries the same idea into everyday practice by bringing design, manufacturing, procurement, quality and the customer into one coordinated process. For small and medium-sized enterprises (SMEs), this integration is both an opportunity and a strain. Their teams are small enough to communicate quickly, but too thinly resourced to run heavy, document-driven processes or to absorb late and expensive redesigns. This review therefore opens with development models, the process blueprints that govern how a firm moves from an idea to a production-ready product, because the barriers, outcomes and design-stage integration discussed in the following sections all play out inside one of these models. Two recent studies from Scopus-indexed engineering journals are examined here. Both start from the same practical observation: agile methods developed for software cannot simply be copied onto physical products, which need early binding decisions, prototypes and test rigs [1, pp. 1–2], [2, pp. 116–117]. Both respond by combining agile practice with a structured, concurrent-engineering backbone rather than replacing one with the other.

Demšar et al. [1] derive their model from a real industrial project. The authors were the mechanical-transmission designers of a pedelec e-bike central drive with polymer gears, which was built by five interlinked sub-teams working as Scrum teams across several partner organisations [1, pp. 5–6, 13]. The project shows how costly rigidity can be. Five weeks after the specification was set, the customer's response to a virtual prototype tightened the targets, raising durability by 10% and cutting weight by 25%. Seventeen weeks later, a competitor's product forced a 100% increase in peak power. Where a traditional process would have gone back to the beginning, the team simply started a new sprint [1, p. 6]. The drive then matured through a sequence of virtual and physical prototypes, from A0 to A3, then B1 to B3, and finally prototype C, which was approved for series production [1, pp. 7–9]. A critical review after the project identified seven weaknesses, including slow response to changing requirements, meetings that were too broad, a poorly maintained backlog, and electronics that could only be tested once the mechanics existed [1, p. 14]. The authors turned these lessons into a general model with three features. The product is split into interdependent modules, each handled by an autonomous but connected Scrum team. Development runs in stages, each made up of several sprints and ending in a functional prototype. Every sprint passes through planning, design, a deliberate decision on whether a physical prototype is needed, virtual or physical testing, and a review

with the customer [1, pp. 15–16]. Alongside this, a five-step accelerated test procedure for polymer gears runs in parallel, so that reliable material data are available before costly injection-moulding tools are made [1, p. 17]. The authors are candid that the evidence comes from a single project, which was also the team's first drive, and that the mix of agile tools must be tailored to each organisation [1, p. 20].

Fargher et al. [2] approach the same problem from the lean side. They built their Lean and Agile Model for Product Development (LAMPD) by mapping a manufacturer's existing Stage-Gate process onto a set-based concurrent engineering (SBCE) model. On top of this they added Scrum for cadence, Kanban for visualisation and work-in-progress limits, and the agile principles as cultural drivers. The result has three gated stages: project definition, project feasibility and project realisation [2, pp. 118–122]. The model was tested over four months at Atlas Copco Henrob, a UK self-piercing riveting site with 200 employees, on a rivet feed-stick used in automotive production lines [2, p. 123]. The model forced the team to explore widely before committing. Value attributes were weighted using the analytic hierarchy process, with reliability at 30% and cost at 25%. The team generated 33 concepts spanning up to 2,640 possible configurations, filtered them using technology readiness level, trade-off curves and design failure mode and effects analysis, and then used a Pugh matrix to select a single configuration [2, pp. 124–128]. The final design has 53% fewer components (nine instead of nineteen) and two fewer sensors. False readings fell from 12% of operations to almost zero, assembly time dropped by about 35%, and decision cycles shrank from three weeks to a few days. The reported savings, about £12,000 in material and about £30,000 from avoided false readings per production line each year, recovered the implementation costs within the first year [2, pp. 129–132]. The study's limitations mirror those of [1]. It covers a single, deliberately low-risk case, the team already knew Scrum, the observation period was short, and the researchers were embedded in the team [2, pp. 132–133]. Taken together, the two studies offer a clear lesson for SMEs. The most promising development model is neither purely plan-driven nor purely agile but a hybrid. A concurrent, gated or set-based backbone secures early knowledge and governance. Short sprints keep the customer close and absorb changes. Virtual prototypes and test-bench data reduce the number of expensive physical iterations [1, pp. 18–20], [2, pp. 131–133]. The decision tools at the centre of [2], such as the Pugh matrix, come directly from the Total Design tradition [3], which suggests that agile hybrids extend that discipline rather than replace it. Whether such models hold up across many SMEs and over longer periods remains to be shown.

References

[1] I. Demšar, B. Černe, J. Tavčar, N. Vukašinović, and D. Zorko, "Agile development of polymer power transmission systems for e-mobility—A novel methodology based on an e-bike drive case study," *Polymers*, vol. 15, no. 1, Art. no. 68, 2023, doi: 10.3390/polym15010068.

[2] G. P. Fargher, M. Aboudaber, T. Bruneau, J. Pandjawatchou, J. Reynaud, M. Regnier, M. Khan, and A. Al-Ashaab, "An industrial application of the lean and agile model for product development: A manufacturing case study," *Concurrent Engineering: Research and Applications*, vol. 34, no. 2, pp. 115–137, 2026, doi: 10.1177/1063293X261423822.

[3] S. Pugh, *Total Design: Integrated Methods for Successful Product Engineering*. Wokingham, U.K.: Addison-Wesley, 1991.