A Model to Predict Span time and Effort for Product Development Processes

Shourav Ahmed[1,2] and Vince Thomson[1]

[1] McGill University, Canada
[2] Bangladesh University of Engineering and Technology, Bangladesh

Abstract. Companies are continuously trying to deliver products on time. However, traditional tools such as product lifecycle management, critical path method, and program evaluation and review technique cannot predict product development span time with reasonable accuracy. Product development involves large, multidisciplinary teams designing complex, interdependent systems; so, predicting span time is challenging. We have modelled product development micro-activities to predict project span time and effort. The model uses a knowledge perspective where the difference between product knowledge requirements and designer knowledge capability drives product development speed. We simulate both process tasks and team behaviour, which are influenced by product complexity, development process complexity, and the difficulty of both technical and interface design. Our method predicts span time and effort at the start of a project or at any point during the development process as conditions change. It can also identify specific bottlenecks and poor levels of designer performance.

Keywords: Project span time, Design effort, Timeliness, Complexity, Product development, Agent-based

1 Background

Many academics have looked into product development (PD) methods in an effort to cut down on product develop time. They have stated that better methods, such as concurrent engineering, agile methods, lean, and systems thinking, have improved PD processes, and that information tools, such as computer aided drawing, computer aided engineering, and product life-cycle management, have improved design time [Wynn and Clarkson, 2018]. These tools have significantly enhanced engineering-in-the-small since small teams can easily coordinate because they utilize the same design tools, communicate easily, and solve problems rapidly. However, engineering-in-the-large where numerous companies and individuals work together, such as in the creation of an aircraft, is still a challenge. The methods used in PD deal with novel creation, uncertainty due to incomplete information, and dynamics where technology changes and customer needs change. Additionally, a significant number of associated tasks frequently rely on data generated by other processes (interdependence). When tasks are interdependent, exchange of information is crucial for timely task completion. It is almost impossible to document and track communication events using a traditional scheduler. Traditional scheduling typically uses a work breakdown structure to identify the effort of individual tasks, and then, optimizes the tasks on the critical path. According to a survey of 211 firms, 55% of PD projects are not delivered on time [Edgett, 2011]. An organization suffers both financial and reputational harm when a product is delivered late. Therefore, timeliness is a vital performance metric for PD processes.

Numerous studies have been conducted to accelerate PD processes; however, faster PD does not guarantee timeliness. The two primary obstacles to achieving timeliness are: (i) methods to estimate span time, and (ii) methods to manage complex PD processes. Researchers have chosen two distinct approaches to resolve these issues: (a) improving scheduling, and (b) improving process flow through the use of suitable coordination mechanisms (organization structure, communication methods, ...). If the process is simple, scheduling techniques such as crashing are effective. However, most scheduling does not work well if the process is complex and has high task interdependency. The development of complex products requires highly specialised resources that are difficult to redeploy effectively, and if there is high task interdependency and if the critical path is dynamic, it is nearly impossible to do better scheduling.

Computer modelling and simulation is the best method for studying dynamic processes. Therefore, a computer-based PD process model was created that incorporated microlevel activities (technical work, interface development, design reviews, and communication) and measured the resulting span time and effort with span time used to estimate timeliness. In the past, lack of computer power hindered the simulation of highly complex processes involving billions of calculations. Recently, computers with high processing speed and the capacity to manage large data sets have been available. The objectives of the proposed research were to investigate the effects of various procedural practises on timeliness, including rapid, efficient, and frequent communication as well as to investigate the effects of macrolevel managerial decisions regarding process procedures and coordination mechanisms.

2 Critical Literature Review and Gap Analysis

Today, in order to stay ahead of the competition, it is not only necessary to provide superior products, but also to bring them to market on time and within budget. Timeliness is highly dependent on the precise estimation of span time, which is difficult to forecast due to task dynamics and process complexity. Traditional methods for estimating span time, such as the Gantt chart, critical path method (CPM), and program evaluation and review technique (PERT), are severely inaccurate [Ballesteros-Pérez et al., 2018]. Gantt charts and CPM cannot account for the many iterations due to rework and are based on estimates of average task time, which underestimates the distribution of possible task times as well as the longest possible task time. In the case of many concurrent tasks, span time should be determined by the duration of the longest task, not the duration of the average task [Savage and Markowitz, 2009].

The complexity of PD tasks makes it difficult to manage and coordinate PD processes effectively. The most significant contributions to complexity are task interdependence, uncertainty, and novel knowledge requirements [Wynn and Clarkson, 2018].

As the complexity of PD increases, so does the need for communication based on knowledge and coordination [Zhang and Thomson, 2019]. McKinsey and Company [1994] found that exceeding the PD budget by 50 percent resulted in a 4 percent profit loss, whereas missing the target date by six months resulted in a 33 percent profit loss. Although researchers have recognized the need for faster PD and for developed methods to achieve it, few have stressed the need to achieve timeliness.

Simulation has proven to be robust and effective in predicting the span time of dynamic and complex PD processes. Among different simulation techniques, discrete event and time-stepping Monte-Carlo approaches have been used most frequently [Wynn et al., 2006; Suss and Thomson, 2012]. One of the significant limitations is that they cannot model microlevel interactions between designers, such as communication, that plays a vital role in PD. A thorough understanding of the inner dynamics of the PD process is essential to developing a method that can predict span time accurately. We need a better understanding of how information exchange, PD process complexity, and organizational structure affect timeliness. Agent-based modelling has proven to be an ideal tool to capture microlevel communication and coordination. However, none of the existing agent-based PD models has focused solely on timely product delivery [Levitt et al., 1999; Zhang and Thomson, 2019]. Most of the mentioned research emphasizes faster product development. However, accelerating a task that is not on the critical path does not affect timeliness. Since it is impossible to track and manage the activities on a dynamic critical path, it is time to try a different approach. We need to investigate what happens when we allow people to organically interact and solve problems that arise at the microlevel.

3 Knowledge Perspective

We wish to model PD as a problem-solving exercise since PD is a learning exercise where many issues have to be resolved. The time required to resolve any problem depends on two variables: the problem's complexity and the solver's expertise. The best characteristic to quantify these two elements is knowledge.

3.1 Knowledge Ability

We are aware that not all tasks demand the same amount of effort to complete. The amount of effort required to complete a task varies depending on a number of variables. How do we evaluate different tasks? Design reviews are used to appraise the quality of completed work. Depending on its quality, a task can either pass a review or be sent for rework. How do we determine a task's quality? These two questions can be answered by modelling the PD process from a knowledge perspective. The knowledge required to complete a task determines the inherent difficulty of that task. Multiple studies have identified the relationship between knowledge, design quality and PD performance [Kim et al., 2013; Markham and Lee, 2014]. A research study compared two methods to estimate span time assessed against actual project durations. Using work breakdown structure resulted in a 27 percent error versus a 13 percent error when using a knowledge perspective [Zhang and Thomson, 2018].

Instead of capturing every possible aspect of knowledge, our model focuses on the technical knowledge needed for designers to complete designs. We categorize technical knowledge as either general or product specific. General knowledge (*GK*) is acquired through formal education or systematic training. For example, the ability to carry out stress analysis on a component can be considered as *GK* for a mechanical engineer. Product knowledge (*PK*) is defined as knowledge related to the development of a particular product that is acquired by past experience. A designer's *PK* anticipates issues regarding systems development and integration, and proactively suggests solutions. Individual tasks create product subsystems. Designers with good *PK* have a high probability of completing designs that can be successfully integrated with other designs to form a system. *PK* is defined on a scale of 0-100% to provide flexibility during simulation.

3.2 Knowledge Requirements (Complexity)

Complexity is the most significant factor that makes the PD process difficult. Complexity is the metric that is used to exhibit the inherent difficulty of a task and is composed of technical complexity and integration complexity. Generally, the amount of effort is proportional to task complexity. From a knowledge perspective designs that require more knowledge (items and levels) are more complex. Designers need *GK* for technical development and *PK* for the integration of functions (interfaces).

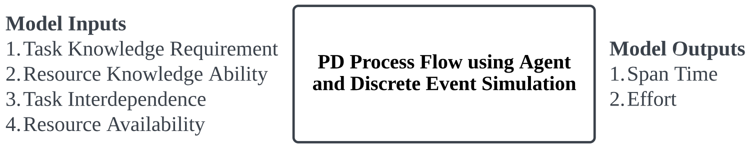

Fig. 1. Black box model of the PD process

4 Case Study: Aircraft Manufacturing

We partnered with an aircraft manufacturer that has a complex product and a very complex PD process due to the large number of subsystems and the large number of partners that have design authority. The aircraft manufacturer uses a stage-gate process for design and development. Our model focuses on a part of the PD process known as the joint definition phase (JDP). In this phase, the manufacturer and its suppliers develop a detailed definition of the aircraft and organize the ownership and development of the required interfaces into functional groups or work packages (WPs) such as structures, systems, power, etc. A single supplier, known as the prime supplier, is responsible for defining the functionality of a subsystem and managing the interface development. Interface management deals with managing communication, coordination, and responsibility distribution between subsystems. Interfaces are managed using interface control documents (ICDs). We consider the development of an ICD as performing a task. Our PD process model determines how project organization, process characteristics, and workflow can improve timeliness.

The aircraft manufacturer uses eight different ICD categories that are required to define all aspects of aircraft specifications. The eight ICD categories are treated as eight process streams. At first, these ICDs are created by designers. Then, they are sent to reviewers by integrators, who manage the PD process, to review the technical aspects of the design. The integrators are experts within the design team who perform an initial quality check and decide the sequence in the case of multiple reviews. Since ICDs impact different aspects of the design, reviews and subsequent approvals take place across multiple WPs. If a design specification is not satisfactory from a technical or interface compatibility perspective, the design is sent for rework. After successful review, the interface design is sent for final approval to approvers. The process flow model of the manufacturer's JDP is shown in Figure 2. The product development process model can be divided into four major tasks: create, review, approve, and rework. These events are discussed in detail in the following subsections.

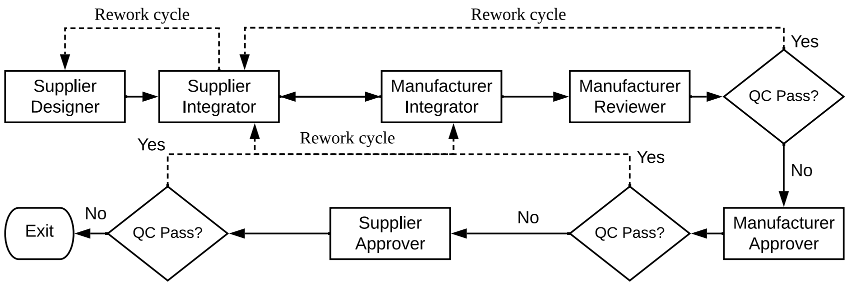

Fig. 2. PD Process flow model of activities for interface development during the JDP

4.1 "Create" Event

The JDP begins when ICDs are created by designers. There are a total of 394 designers spread across the 8 different ICDs. Designers mostly belong to supplier WPs. The creation of all 8 ICD types starts simultaneously as they use separate resource pools. It is important to note that designers not only create the ICDs, but also modify or correct them during rework. Therefore, there are instances when all designers are busy. If no designer is available, ICDs wait in a queue. We used ICDs as agents to store information related to knowledge requirements and capabilities. Whereas knowledge requirements are predetermined for each ICD, designer capability (knowledge level) is stored inside the ICD agent only after a designer is assigned to a task. This information is carried inside the agent for subsequent operations and is used in judging ICD quality. This information is stored as variables that can be changed dynamically during the simulation. For example, the knowledge capability of a designer increases through consultation with an expert. The ICD managed by a designer is updated automatically when a designer's new knowledge capability changes, as it is a major determinant of quality. A triangular distribution of effort in hours was obtained from the manufacturer and used to calculate effort. The effort to create an ICD is proportional to the knowledge difference between a task's knowledge requirement and a designer's capability. ICDs are then sent to reviewers via integrators who manage the process flow. Usually, there are integrators both on the supplier side and the manufacturer side. The supplier integrator first verifies the quality of the ICD before sending it to the manufacturer integrator. The manufacturer integrator decides the sequence of reviews by one or multiple reviewers.

4.2 “Review” Event

Reviewers check the quality of an ICD from both a technical and interface perspective. Due to the multidisciplinary nature of the ICDs, reviewers from multiple WPs need to verify the quality of an ICD from their perspectives. There are 206 reviewers in total. Some of the reviews are independent, while others are interdependent. A review is called independent when a reviewer from a WP is responsible for reviewing a single ICD type. The probability of passing a technical review is directly proportional to the general knowledge capability of a designer and inversely proportional to the technical complexity of the ICD. After a review is completed, a decision is made about whether an ICD should proceed forward or be sent for rework. In the case of interdependent reviews, reviewers often need to collaborate to ensure a consistent interface.

The need for collaboration depends on the integration complexity of the ICD. The higher the integration complexity, the higher the chance of requiring collaboration. Collaboration is done to ensure a consistent interface specification. There are generally two types of collaboration: vertical and horizontal. Vertical collaborations are done by reviewers from the same WP who work on different ICD types. Vertical collaboration meetings are held in pairs. Two reviewers from the same WP and working on two different ICDs meet and come to an agreement regarding the overlapping aspects of the design. Pairwise meetings continue until every reviewer is updated about the work of other reviewers in the same WP. In horizontal collaboration, all reviewers across multiple WPs and the designer in charge of the ICD meet to ensure all aspects of the design are satisfied. Reviewers from different WPs sometimes have conflicting objectives when making a design choice. For example, in choosing the material of a major component, the structural engineer would be more biased toward selecting a heavier material with high strength and rigidity. On the other hand, the engineer in charge of weights would be biased towards selecting a lighter material that would enable a higher payload. These collaboration meetings allow reviewers to discuss these differences and select the best design alternative. After collaboration, the reviewers decide if the ICD is consistent with other ICDs from a product knowledge perspective. The probability of passing this quality test is equal to the designer’s product knowledge, which has a value between 0 and 1. The rationale behind this is that the greater the product knowledge of the designer who created an ICD, the better the chance of creating a consistent interface.

4.3 “Approve” Event

The final step in the JDP process is approval. During the approval process, an approver ensures that all the changes suggested in the previous steps are implemented properly. No collaboration is required for this step. The approvers from different WPs work independently to ensure that the design specifications are consistent from their point of view. The amount of effort it takes to approve an ICD is calculated in a similar way to calculating review effort. The probability of passing a quality test is equal to the designer's level of product knowledge. The JDP is considered complete when ICDs from all 8 ICD types are approved.

4.4 “Rework” Event

Rework begins when an ICD fails a quality test at any point during the JDP. The designer with the highest level of technical knowledge is selected from the resource pool that created the ICD. Following rework, the ICD has a better chance of passing the quality test since the probability of passing is equal to the higher level of product knowledge of the new designer.

However, there are times when the designer from the pool with the most technical ability knows less than the designer who previously worked on an ICD. A consultation with an expert is needed in these cases. During a consultation event, the designer with lower knowledge meets with an expert who belongs to the same knowledge domain and has a significantly higher knowledge level. Since the number of experts is limited, the designer must wait for an expert before starting a consultation. The increase in knowledge after consultation is proportional to the knowledge gap between the expert and the designer as well as to the duration of the consultation. Missed consultation events result in significant wasted effort that increase span time and total effort. Reworked ICDs after consultation are sent to integrators to continue the JDP process.

5 Results

In our case study, we tested different scenarios and compared them with a baseline scenario. Each scenario was run 100 times with a different initial seed to account for the randomness in the probabilistic input data. For the baseline scenario, the span time was 1729 ± 213 hours and the total effort was 345,000 ± 5,500 hours. On average, the standard deviation of span time was 12% and the standard deviation of effort was 2%. All results are relative in nature. We investigated the effects on span and effort for several scenarios using data from the case study. A few of them are discussed below.

5.1 Varying Task complexity

We investigated the impact of varying task complexity on span and effort. The use of new and improved technology is a good example of higher technical complexity. We tested 5 cases by lowering or increasing complexity in increments of 10% (Fig.3). For each 10% rise in complexity, there was a 6% increase in span time and a 4% increase in total project effort.

5.2 Varying Designer Capability

We investigated the effects of a more mature work force on productivity, time, and effort. We increased designer knowledge level by 20% compared to the baseline knowledge level of 100% and we observed about a 2% reduction in span time and about a 6% reduction in total effort for each incremental change in knowledge (Fig. 4).

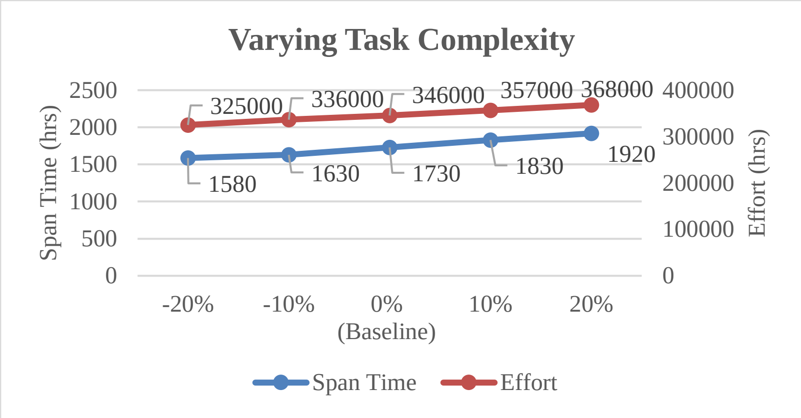

Fig. 3. Effect of varying task complexity on span time and effort.

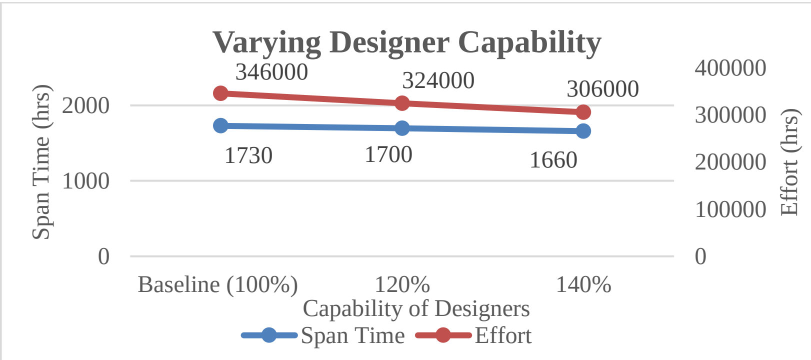

Fig. 4. Effect of varying designer capability on span time and effort.

5.3 Delayed Start of a Bottleneck Resource

Negotiations can often result in the delayed start of a resource. If it is a bottleneck resource, the effects on span time and effort are disastrous. We tested 2 delay points (3 and 6 months) and compared them with the baseline for span time. We observed that, for a 3-month delay, the span time increased by about 9.3% from 1730 hours to 1890 hours (Fig. 5). For a 6-month delay, the span time was 2140 hours, an increase of 23% from the baseline. Total effort was unchanged as the delayed start did not add work.

5.4 Adding More Resources to Bottleneck Tasks

Finally, we wanted to study the effect of adding more resources to bottleneck tasks at different points, such as the beginning or midpoint of the PD process, compared to the baseline case. The span time decreased by about 24% when we added 20% more resources at the beginning (Fig.6). When we did the same at the halfway point, the improvement dropped to 18%. There was no significant change in total effort because the amount of work in all three test cases remained constant.

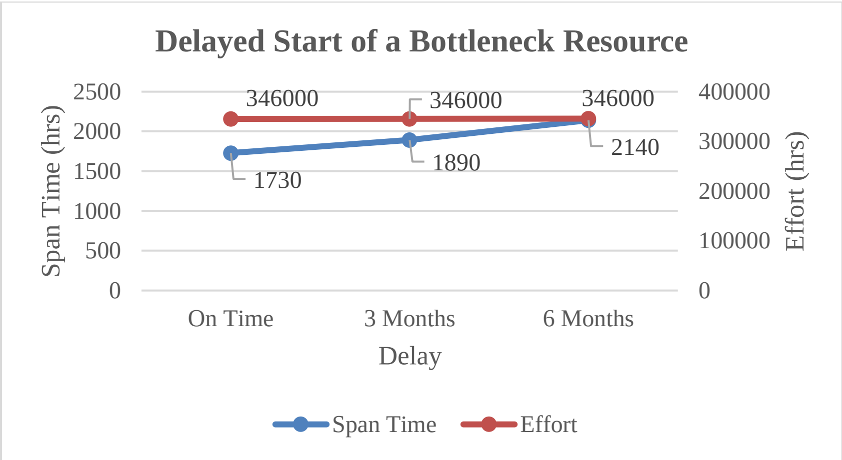

Fig. 5. Effect of delayed start of a bottleneck resource on span time and effort.

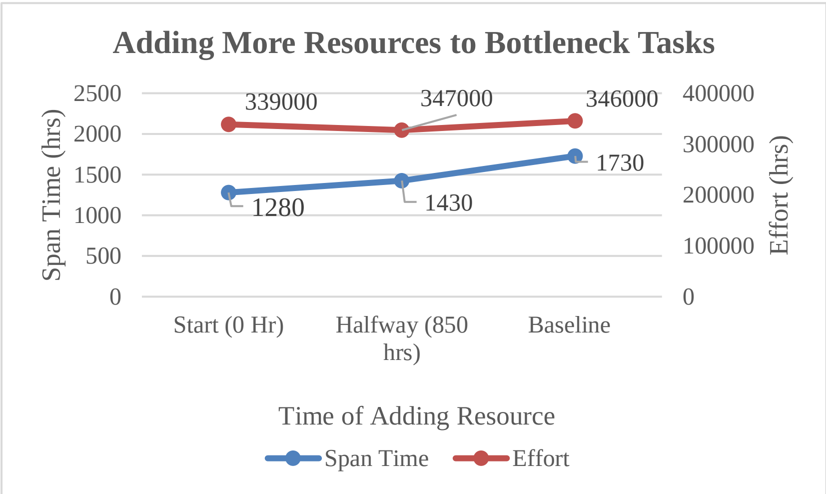

Fig. 6. Effect of adding more resources to bottleneck tasks on span time and effort.

6 Conclusion

Modelling PD processes using a knowledge perspective worked well. Simulations of an actual PD process gave results that were consistent in the estimation of span time and effort for variations of different process characteristics. We found that increasing task complexity increases both span time and effort, and increasing designer capability reduces both span time and effort. We also learned that delayed start of a bottleneck resource increases span time drastically.

The knowledge perspective model was able to estimate span time and effort, and thus, on time delivery of PD for complex, dynamic processes. The model can help managers to determine the impact of decisions on the timeliness of project outcomes. Managers can estimate the effect of technology (complexity) and resource maturity (capability) on future projects. The methodology will work on any type of project where knowledge is an appropriate characteristic to model the difference between product requirements and resource capability. In the future, we would like to extend the methodology to the entire PD process. This method can also be used for other coordination mechanisms such as agile methods.

References

1. Wynn, D.C., and Clarkson, P.J., Process models in design and development. Research in Engineering Design 29(2), 161-202 (2018).
2. Edgett, S.J., New product development: Process benchmarks and performance metrics. Stage-Gate International (2011).
3. Ballesteros-Pérez, P., Larsen, G.D. and González-Cruz, M.C., Do projects really end late? On the shortcomings of the classical scheduling techniques. JOTSE: Journal of technology and science education 8(1), 17-33 (2018).
4. Savage, S.L. and Markowitz, H.M., The flaw of averages: Why we underestimate risk in the face of uncertainty. John Wiley & Sons. (2009).
5. Zhang, X. and Thomson, V., Modelling the development of complex products using a knowledge perspective. Research in Engineering Design 30(2), 203-226 (2019).
6. McKinsey and Company, Lead local, compete global: unlocking the growth potential of Australia's regions. McKinsey (1994).
7. Wynn, D.C., Eckert, C.M. and Clarkson, P.J., Applied signposting: a modeling framework to support design process improvement. In: ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. 553-562. American Society of Mechanical Engineers Digital Collection (2006).
8. Suss, S. and Thomson, V., Optimal design processes under uncertainty and reciprocal dependency. Journal of Engineering Design 23(10-11), 829-851 (2012).
9. Levitt, R.E., Thomsen, J., Christiansen, T.R., Kunz, J.C., Jin, Y. and Nass, C., Simulating project work processes and organizations: Toward a micro-contingency theory of organizational design. Management Science 45(11), 1479-1495 (1999).
10. Kim, T. T., and Lee, G., Hospitality employee knowledge-sharing behaviors in the relationship between goal orientations and service innovative behavior. International journal of hospitality management 34, 324-337 (2013).
11. Markham, S. K., and Lee, H. Marriage and family therapy in NPD teams: Effects of We-ness on knowledge sharing and product performance. Journal of Product Innovation Management 31(6), 1291-1311 (2014).
12. Zhang, X., and Thomson, V. A knowledge-based measure of product complexity, Computers & Industrial Engineering, 115, 80-87 (2018).