Proceedings of the ASME 2014 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2014
August 17-20, 2014, Buffalo, New York, USA

DRAFT DETC2014-34275

DESIGN MARGINS AS A KEY TO UNDERSTANDING DESIGN ITERATION

Claudia M. Eckert
The Open University
Milton Keynes, MK7 6AA,
United Kingdom

Ola Isaksson
GKN Aerospace Engine
Systems, 461 81 Trollhättan
Sweden

Chris F. Earl
The Open University
Milton Keynes, MK7 6AA,
United Kingdom

ABSTRACT
Design processes are subject to many uncertainties. Changes resulting from the need to respond to external uncertainties are one of the main drivers of engineering change and therefore for iteration in design processes. Another important cause of iteration in design processes arises from the dependencies in design information which is being generated as part of the design process itself. At the beginning of the design process engineers need to make an informed guess about the values of parameters that they need and can achieve. These values are passed on to others, who base their decisions on them. Design decisions are distributed and iterative among design teams, customers and suppliers. Communicated parameter values are uncertain in two different but related ways. First, there is the confidence, precision and commitment that the designers have in the values they specify. Second there are uncertainties in the values that can be achieved with the technology the new design employs. These issues become particularly challenging when they span design teams, customers and suppliers as they iterate to converge on a mutually effective solution. This paper looks at this type of convergent iteration through an example from the aerospace industry, which illustrates how uncertainty in operating temperature at the beginning of the design process requires a thorough understanding of the temperature ranges that solution alternatives, at different degrees of maturity, can operate under. This paper argues that the key to managing convergent iterations lies in communicating the available ranges of parameter values and in understanding how design margins have arisen in existing technologies. These margins on product parameters provide potential performance which exceeds immediate functional requirements. The paper develops and formalizes the concept of design margins and argues that margins are included into products for a variety of reasons that are not always transparent to different team members. Analysis of margins enables design companies to reason in terms of ranges of values describing the scope for design change in meeting customer and supplier requirements without being forced into unplanned iteration loops.

INTRODUCTION
The design of complex products inevitably involves iteration. Tasks have to be repeated or their results adjusted because of new information arising from other tasks or external factors. Understanding how design iterations work is critical to effective planning of design processes. Too little time planned for iteration can put delivery dates at risk, too much time can look to a suboptimal use of resources and unnecessary cost. However, iteration does not only affect process planning, but also fundamental decision making about how a product is designed. In early stages of design projects fundamental decisions are made about the ability of existing solutions and solution principles to meet new requirements and therefore also about the need to deploy new approaches. In complex products many such fundamental decisions have to be made, each associated with a degree of uncertainty. In some cases the existing technology is well up to meeting the new requirements and can be deployed as it is, in other cases it will need to be modified or a new solution will need to developed. For the overall product some solutions will be reusable and others will have to be redesigned. However, which one will need to be changed is difficult to judge at the beginning, because of the complex interaction of different parts of the product. For example in jet engines the aim is to achieve as high combustion temperatures as possible to optimize fuel consumption. What exact temperature can be achieved depends how the cooling design and the material properties interact in particular

configurations. However, operating temperature is a fundamental parameter for many other components in the jet engine. Therefore the designers of these components have to start working on their components, not quite knowing what temperature they will need to accommodate. As the final temperature becomes clearer, the other parts will have to be redesigned, unless they could operate under a much higher temperature. This implies a risk of overdesigning a component, which could be costly. Convergent iteration is typical in the design processes of many complex products [1].

This paper unpacks the iteration resulting from information interdependency in terms of the margins in existing or potential solutions. This allows aspects of the dynamic nature of design processes to be explained. The limits of one component's or system's ability to meet certain requirements may be lower than anticipated. Then other parts of the system need to compensate. Alternatively limits might be higher and designers will want to take advantage. Unlike other approaches which address the structure of the links and the flow of information (see for example [2]), this paper argues that understanding design margins is critical to convergent design problems, where a certain degree of iteration is inevitable. However, assumptions about the values reached by others and capabilities can lead to unnecessary iterations. If the realistic values are only recognized once the design is frozen this may result in potentially costly engineering changes rather than being resolved as part of iteration. The problem is twofold: the margins are not known to the designers and the designers are unwilling to communicate vague and uncertain information early in particular across the supply chain. Communicating such information may risk being misleading and potentially compromise business relations.

This paper combines a theoretical analysis of the concept of design margins with the results of a series of interviews that we have conducted in a Swedish truck manufacturer [2]. It also draws on the second author's experience in jet engine component design teams.

After a brief discussion of research background, the paper introduces an aerospace example which illustrates the role of margins in design decision making and planning. This leads to an in-depth discussion of the concept of component and system margins. Returning to the example, the paper argues that companies can gain from making an explicit effort to understand and record component margins as well as communicating this understanding clearly among parties in the design process

THE DESIGN PROBLEM

Jet engine design is "contract based", which means that business agreements exist among developing partners and customer orders have been placed in advance of the product development work. Conceptual Design (where configurational options have been studied) has been undertaken in pre-development activities. Requirement specifications exist only at a comparatively high level at the stage where development partners engage in development activities. Necessary information required to detail the engineering solutions such as the loads, interface positions etc. evolve during the development process. It is common that there is a process defined in the development and verification plans of partners for how versions of the product solutions and the various loads and interfaces are being exchanged and updated.

As design decisions are being made, the engine integrator and various component/sub system development teams meet to converge the product solutions. The allowable design variation of the design reduces. The concept is being refined. Initially – architectural options are being studied, where alternative topologies, materials, and manufacturing approaches are explored. At this stage the knowledge about the final constraints, loads etc. are based on a first set of predictions of the desired behavior and the governing high level requirements (target weight, performance parameters and restraints in cost dimensions etc.). As the design strategy has been chosen, the levels of detail required to make further decisions (sizing, dimensioning, optimization etc) require more detailed understanding of the loads, restraints and dependencies with other components. Based on what component technology is selected – the behavior of the overall system is affected, which in turn result in governing conditions (loads) for the component. Although even the initial requirements of the engine states expected life length (maintenance intervals etc), the information to design against such requirements depend on both a careful understanding of the load cycle, originating from the flight cycles, and certification load cases (failure modes). Such governing information depends on how well the engine components behave together as an integrated engine, and analyses on an integrated level often results in updates on the detailed behavior of the engine, such as the thermal gas load during various conditions.

The companies have each developed different solution approaches to certain technology readiness levels, and need to make a choice which technology to pick at the beginning of the process making assumptions about the capability of their own technology and that of their partners. Technology Readiness Level is commonly used in the aerospace and aero engine business to assess how mature a certain technology is. The TRL scale originates from NASA [4]. It ranges from TRL 1 (Basic Principles observed and reported) to TRL 9, for a well-established maturity level – a Standard solution. TRL 6 is generally required for introduction in a product development process. One basic principle of the TRL scale is that any technology is mature only within in its specific context. Therefore a technology which has been assessed at TRL level 7 for a certain engine application can drop in maturity level when applied in another engine application. To climb above TRL 4 requires validation in a relevant context, which may require rig tests and subsequently flight tests. A novel technology, or a technology used in a new context, requires validation work to reach higher TRL levels. This typically drives cost and lead time that may both be scarce resources in a development program. Therefore companies aim to use technologies of high readiness levels unless they want to push a particular technology, knowing they will need it in the future. Manufacturing companies also uses other maturity metrics, such as manufacturing readiness levels.

Planning the design activities for a company is a matter of assessing the risks of changes from the customer as well as an understanding of their own ability. This is further complicated by the way in which this uncertain information is communicated throughout the development process. Uncertainty leads to a delay in actually communicating information. Rather than releasing imprecise information and the range of uncertainty associated with it, companies wait until they are surer to avoid unnecessary iteration due to imprecise information. However, this occurs at the risk of reducing the development time available for the suppliers. If the development does go on in parallel this can leading to churn [4], where projects appear to be on time and then spilt in their schedules If the suppliers had an idea of the range of expected values they could identify technologies for worst case scenarios. By communicating crisp values the costumers hide both their commitment to particular values and the confidence that these will either be met or exceeded. In a culture where convergent iteration is the norm because it is difficult to anticipate how the exact performance values will be progressing, it is important to communicate clearly the likely changes to the values. The company and its customer are sharing information about the TLRs, which affect product risk in use, rather than information about the performance parameters, which would be useful mainly for planning design processes.

In this paper we will explore the relationship between design margins and iteration. TLR can be increased by increasing design margins, or by increasing the understanding of design margins. Communicating design margins would be one way of providing greater transparency for all parties.

BACKGROUND

This industrial example illustrated the challenges of needing to reduce iteration and carrying out effective changes at each stage, which excessive propagation. These problems are aggravated by the companies' reluctance to communicate uncertain information.

Iteration

Iteration is recognized as a fundamental part of most design processes and their models [6], either seen as part of exploration and convergence or as feedback loops in a stage-based process. Iteration is also seen as something to be minimized or avoided by defining design problems more clearly or confining iteration to the early phases of design processes [7] in what Safoutin [8] calls the "zero-iteration ideal". Planned design iterations, however, are a natural and important means for designers and engineers to gain understanding of the design, especially coupled problems. For example gas temperature in a jet engine critically influences the dimensioning of components. Conversely, gas temperature is determined by the component details arising from integration into the whole product. Several iterations are required to converge to a solution.

This difference of perspective is not surprising considering that iteration has a range of roles and manifestations in the design process. Wynn et al. [1] distinguish six non-orthogonal perspectives.

- Exploration involving a repeated process of solution space divergence (during synthesis) followed by convergence (during evaluation).
- Convergence, where parallel tasks are repeated as parameters and objectives are brought in line to converge upon a 'satisficing' design. Design Churn, defined by [4] as a scenario where the total number of problems being solved does not reduce as the project evolves is typical during a design convergence.
- Refinement, where a design or a part of it is reworked to meet slightly modified or secondary requirements.
- Rework, where a task is revisited in response to an emerging problem.
- Negotiation, where trade-offs are made to meet conflicting goals often across disciplinary boundaries.
- Repetition, where similar tasks or steps are performed at different points in the design cycle or to meet different goals.

Other researchers ([9],[10]) argue that iteration occurs, because new information becomes available or new goals need to be addressed. Design Structure Matrix (DSM) [11] approaches describe iteration as cyclic dependencies in design processes or system architecture and classify iteration by the relationships between tasks. Cho and Eppinger [12] distinguish three types of iteration. First, in overlapping iteration, tasks need to release information prematurely for other tasks to start, which than have to be revised when refined information is available. Second, in sequential iteration, downstream tasks modify information used by an earlier task so that potentially all intermediary tasks have to be readdressed. Finally, in parallel iteration concurrent tasks depend on on-going information exchange.

Parallel iteration is the mechanism through which convergent iteration takes places. In a DSM parallel iteration can be identified through sub-matrices with lower and upper diagonal dependencies. Smith and Eppinger [13] define a strength dependency measure and calculate the eigenvectors and eigenvalues of these strength dependency matrices to determine the rate and nature of convergence in design processes to identify the relative importance of individual tasks in parallel iteration. As many tasks are not carried out completely in parallel, but overlap, planning when to start a specific task is critical [13].. Rather than addressing these structural issues, this paper looks at iteration from the perspective of the individual elements being able to meet the expected values with which the parallel iteration cycles start; and therefore their propensity to force other components to change or push a modification of the requirements.

Engineering change

Iteration and engineering change follow similar processes, but start from a different degree of commitment to values. Freezing critical design dimensions, such as interfaces is a commonly used mechanism to control (even contractually) the refinement and progression of products [16]. In practice freezing geometric properties is easier than freezing non-geometric properties, such as loads, that are harder to control because they are less localized. Freezes are a mechanism through which parallel iteration can be broken up into smaller clusters. As some components cannot be altered, for example they have been ordered with long lead times, others have to adapt to them. If frozen components have to be modified, an iteration becomes an engineering change. Engineering change is usually defined as “an alteration made to parts, drawings or software that have already been released during the product design process. The change can be of any size or type; the change can involve any number of people and take any length of time.” [17]. By contrast iterations generally occur before a design element has been released. A change process is therefore more formal than a design iteration. It needs to be approved and signed off. Changes are difficult to manage, because it can be hard to foresee when changes will occur and what will be affected by a change. Changes can lead to unexpected knock-on effects in other parts of the product. If the changes snowball or avalanche this can pose a serious risk to process plans and budgets, making it difficult to plan the resources required to deal with the change [18]. If the resources are not available the need for change must be resolved in a different way. For example a change that was intended to be carried out internally, might be passed on to a supplier, or the resources pulled from other design tasks. Therefore the effect of changes can spread across organisations [19].

Each proposed change can be expressed as changes to one or more parameters of the component or system. Each component is associated with various parameters, which might become critical in the sense that small, required changes (or combinations of changes) initiate significant changes in components with linked parameters (Figure 1).

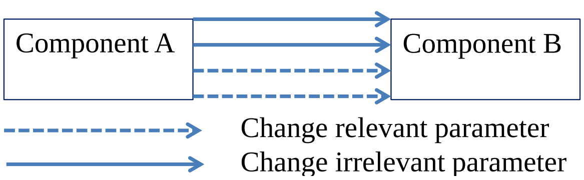

Figure 1 Parameter links between components [18]

Components either act as (i) change absorbers, receiving change without passing it on, (ii) as change carriers passing the same degree of change on as they have received, or (iii) as change multipliers, which pass more changes on to others parts of the system, which will in turn need to be redesigned [18]. Change becomes problematic when change absorbers turn into change carriers or change multipliers and critical when changes avalanche creating more and more potential changes. Often the changes can be brought under control given sufficient design resources and time, but companies might need to abandon these projects. From the perspective of an individual change, there might be several knock-on effects some that can be dealt with directly and others that require detailed analysis. This leads to ripple effects with the volume of change decreasing over time. Griffin et al. [20] has shown that typical patterns of change propagation can be identified. Pasqual and de Weck [21] studied change requests for a complex product over many years, observing ripple patterns, with the number of change requests increasing towards deadlines.

Change is difficult to predict, because it is inherently not deterministic. Designers need to do several things; make explicit decisions about how to implement a change [22], understand product connectivity [23] and understand the ability of the component to absorb a specific change [18]. Changes do not freely propagate through all the potential connections of a network, but are highly constrained. Various tools have been developed to predict the impact of change, through probabilistic links [24] anticipating changes [25] or underlying functional models [26]. More recently change prediction has been integrated with visualising other aspects of the product [27] and integrating product based change prediction with process models [27], [28].

Communication and collaboration

Convergent iteration in the design of complex product usually involves the coordination of groups of people. While this might occur in some cases within collocated teams who can exchange information easily in an *ad hoc* manner, the parallel tasks are often carried out by different teams. Therefore the challenges arising from the inherent task dependency are often increased by the need to communicate the information across expertise or organisational boundaries [30]. To achieve effective communication the designers need to achieve a certain degree of shared understanding of the product and each other’s tasks [31] and find representations that can be understood by the different groups [32]. Convergent iteration can be problematic across different companies in the supply chain, who introduce the risk of time delays and cultural misunderstanding while being susceptible to the effects of power relations in the supply chain [33]. The customer or original goods manufacturer aims to maximise their own advantage in terms of performance, design time or cost. The customer can try to force changes into a supplier instead of changing their own design, i.e. instead of a parallel iteration the iteration is pushed to the supplier’s tasks. Alternatively the supplier can also demand changes from their customers in response to change to their components. Customers also often approach multiple potential suppliers and the supplier runs a real risk of losing the order, if they cannot meet the customer's demands.

When convergent iteration cycle begins each party has a vague idea of the design they plan to create and the performance values that they can achieve and communicate these values to others. The designers themselves might have an idea of the range of values that they can achieve, but are often reluctant to disclose them, because they don’t want to mislead others. With the consequence of either communicating crisp values early, which later are changed or the hold back with communicating values until they are fairly sure and thereby reducing the design time available for their colleagues. The uncertainty associated with seemingly crisp design values can be classified into the following categories [34]: precision typicality, commitment (the opposite of provisionality), sensitivity input confidence, understanding and confidence. These values can be seen as margins on the communicated values of a parameter, rather than the actual value of that parameter.

DESIGN MARGINS

The ability of a component to absorb or multiple a change is at the heart of managing both iterative product development processes and engineering change. This ability comes from the margins of a component with regards to specific functional requirements. This section introduces the concept of design margins, which is then used to explain the iteration in the industrial example in the following section.

Definition of margins

In Eckert et al. [35] a design margin is defined as "the extent to which a parameter value exceeds what it needs to meet its functional requirements regardless of the motivation for which the margin was included". For example if a beam is required to carry a weight of 5 kg, but could in fact carry 7 kg than it has a margin of 2 kg. However each parameter can also be subject to constraints and if these are exceeded the design is no longer viable. For example if the beam is sitting in brackets at either end that can only carry 6 kg, the margins exceeds the constraints on the component, as illustrated in Figure 2. The base line parameters that are required are drawn as a heavy black outline and the actual component parameters are shown by the grey shape. The constraints are expressed through the dashed lines. Viable margined products sit within constraints.

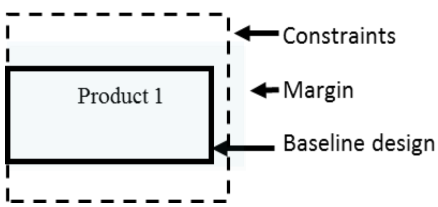

Figure 2 Requirements, margins and constraints [35]

Most components and systems have to meet a multitude of different functional requirements. In the following we use the term 'product' as a generic term for component, subsystem or the whole product. A product meets several functional requirements, fr_1, fr_2 ..., which may be aggregate sets of requirements associated with a particular type of performance. The set of functional requirements may be represented as a requirements vector

$$FR = < fr_1, fr_2, \ldots fr_n > \qquad (1)$$

The term product is used as a generic term for component, subsystem or whole product. Individual functional requirements can correspond to components and subsystems. In the context of iteration these can in turn correspond to functional design teams.

For each product functional requirement there are a set of product parameters which determine the performance of the product with respect to meeting fr_i. Let these be denoted

$$\boldsymbol{x}^i = < x_1^i, x_2^i, \ldots x_p^i > \qquad (2)$$

Parameters may be common across the functional requirements $\{fr_i\}$. For iterations in complex product development this can present difficult issues. One functional team focuses on parameter set $\boldsymbol{x}_i$ to deliver its functional requirements while another deals with an overlapping set $\boldsymbol{x}_j$.

The mapping between the parameters and functional requirements is identified as performance P. Assuming that it is possible to evaluate the performance P precisely. Then design parameters $\tilde{\boldsymbol{x}}^i$ which deliver requirements exactly satisfy

$$P(\tilde{\boldsymbol{x}}^i) = fr_i \qquad (3)$$

However, both assumptions are ideal. The evaluation of performance is inherently uncertain and for a given design proposal the exact functional requirements may be missed, either exceeding or falling short. Product margins arise in the case when a particular set of parameters has performance which exceeds the functional requirement. However, initial design analysis is simplified if P is monotonic in each parameter as defined by Papalambros [36]. It is assumed that P is monotonic increasing in each parameter, so that parameter margins for an actual design with parameters, $\boldsymbol{x}^i = < x_1^i, x_2^i, \ldots x_p^i >$ if they exist, are

$$\boldsymbol{x}^i - \tilde{\boldsymbol{x}}^i = < x_1^i - \tilde{x}_1{}^i, x_2^i - \tilde{x}_2{}^i \ldots x_p^i - \tilde{x}_p{}^i > \qquad (4)$$

The corresponding performance margins are

$$P(\boldsymbol{x}^i) - P(\tilde{\boldsymbol{x}}^i) \qquad (5)$$

There may be multiple solutions to equation (2) and corresponding multiple margins. Tradeoffs with additional requirements or preferences may identify one set of parameter margins. Further, and more critically, common parameters may be associated with distinct functional requirements. That is parameter sets

$\boldsymbol{x}^i = < x_1^i, x_2^i, \ldots x_p^i >$ and $\boldsymbol{x}^j = < x_1^j, x_2^j, \ldots x_q^j >$ for separate functional requirements may share parameters. In this case one functional requirement may dominate a parameter, forcing the higher value to meet equation (2) and creating an apparent margin with respect to the other functional requirement. For separate teams working on these different functional requirements sharing of parameters presents a critical communication issue for coordinating iterations.

At any stage of product development and design the evaluation of performance is more or less uncertain, depending on the maturity of design proposal as well as the resources and tools employed in evaluating the proposal. These uncertainties give rise to margins of different type as allowances made in parameters to provide high probabilities of meeting functional requirements. In some cases the functional requirements themselves may be uncertain with expected and possible requirements articulated. In these circumstances parameters and functional requirements are in principle expressed as probability distributions and in practice as upper limits on requirements and corresponding parameters. However, a key point to note is that although design margins can exist above these limits, margins can also be created or increased by reducing uncertainties, for example by better models, more exhaustive simulations or physical testing of components and prototypes. So product parameters may have 'safety margins' which are not immediately usable in design but offer the potential that they can be reduced and release tangible and usable margins. These offer options for meeting higher levels of functional requirement or optimizing product parameters against existing functional requirements. However, the reality lies in between. Functional requirements 'increase' taking up some of the parameter margins released but at the same time some of the newly available margins are maintained allowing for further future evolution of functional requirements.

Another scenario occurs when one parameter is 'frozen' and the design objective is to maximize performance by varying other parameters within their limits. For example maximising performance from a newly developed material.

The mathematical properties of the performance map $P(\boldsymbol{x}^i)$ will be critical. If very small changes in $\boldsymbol{x}^i$ give large changes in $P(\boldsymbol{x}^i)$ redesign near the margins will appear risky, especially with the inherent uncertainties in the map $P(\boldsymbol{x}^i)$. There are continual trade-offs between parameters, both material and form, and performance against several functional requirements as iterations take place. Margins express the room for maneuver in these trade-offs.

Product performance is the measure of function and behaviour, i.e. how well the product performs what it is designed to do [36]. Margins fall into two categories performance margins, and parameter margins, typically geometric and material, which describe how much the form or material properties of a product or component can be changed before its performance is affected. An important type of geometric parameter margins are clearances surrounding a component. These allow the component to grow or to be moved and can therefore buffer potential changes to the product. Clearances can be difficult to handle, because they are not explicitly represented with a particular component but rather deduced from relationships between the components.

Margins in the design process

While the functional requirements of a product might be known early in a design process, the design margins change through the process. Components or systems are often initially designed with a certain room for growth in mind. For example the beam might be designed to carry 7 kg, even though only 5kg are initially required, because the designers assume from experience that that weight will go up as the design process progresses. Figure 3 illustrates how the margins might be eroded as the design processes.

The beam is a simple example where one component meets one functional requirement. In practice many functional requirements are met up multiple components and the exact way these components interact depends exactly what the margins are, for example the structural stability of an aircraft wing is (amongst other factors) resulting from the geometric arrangement of the struts. Each strut on its own can contribute a certain amount to the stability, if this is not sufficient than more struts needs to be added or a different material is selected. The exact details of how much each component can contribute to a functional requirement only become clear as the component is being designed. For example the contribution of a stiffener depends on the overall wing geometry, but also on detailed design decision, such as the way in the components of the wing are connected.

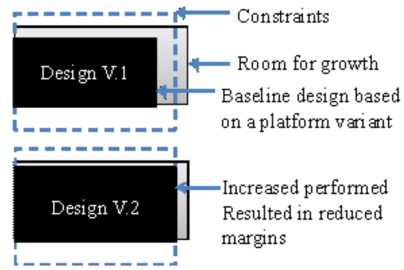

Figure 3 Reduced margins of change

Knowing margins

While a component margin is an intuitive concept, it is in practice very difficult to know what the margin on a component actually is. For a finished product, margins can be established through testing or potentially through simulation. However, in current industrial practice most products are tested to establish that they are capable of meeting of the specified requirements in the specified way [38]. The actual margins of the component can be revealed through expensive destructive testing. Many margins can be elicited through computer simulation.

The margins on a component under development cannot be established accurately, but might be described probabilistically. Typically designers of a complex product start from an existing design and their understanding of its margins with respect to the new requirements, which can be informal or the result of testing. While companies are usually not aware of all component margins, they often have a picture of the component margins that have become critical in a previous project. For example, if a clearance has nearly been used up and a real effort was needed to fit everything in, for example when fitting cables through a clearance, they know that it is very unlikely that they can push this component (or clearance) further. This gives designers a picture of the high risk areas. At the beginning of a design process the designers therefore access the previous design as to where they can make improvements and which parts of the product might be overdesigned and therefore could absorb higher requirements. They know that margins on components can absorb change, while margins that are exceeded become change multipliers. However, often different teams look simultaneously at the possibility of change in a product and target the same components or areas. For example the electronics team identifies a clearance for another cable and the design team is planning to increase the size of an adjacent component and wants to fill up this space. This might be picked up at a formal design review, but between reviews it requires somebody to stop the potential problem. This could therefore lead to unintentional rework, due to problems with the design.

The design teams are also not clear at the beginning of the design whether are able to reach or exceed the intended performance requirements as we will discuss in more detailed with the following industrial example.

Ways of thinking about margins in an organization

Another reason why it is difficult for designers to know the margins on a component, is that different groups in an organization think about margins in different ways and in consequence use different words to talk about the same underlying phenomenon. Designs teams may add *safety margins* to the requirements of the components of the product against emerging requirements and uncertainties during the development process. Companies are however, not just thinking about the current product, but often have future generations of the product. They aim to redesign major systems every few generations and therefore need to plan in margins for the predicted requirements of future products, for example the current system might need 12 electronic ports, but the company excepts that each generation adds 2 ports and therefore designs system with 16 ports, so that they don't have to redesign the ports for another two generations. The *room for growth* arising from product planning comes at a cost, because the current system has an overdesigned system. However, not only future generations of the product might lead to overdesign, but also versions of the same product. Therefore there is a real tension between the optimization of an individual product and the flexibility which margins give an organization (see [5]).

Overdesign is also the way the margins are described from a value engineering perspective. For example the beam that can carry 7kg rather than the required 5kg might be heavier or more expensive than one that actually only carries 5kg. Therefore the component might be targeted in cost saving activities, for example when a company is thinking of bringing out a cheaper product that is targeting another market. Overdesign is also considered when companies are trying to synchronize service intervals on a product. For example jet engines are taken off the wing at regular intervals and any component is replaced that is likely to cause problems during the next service intervals. Phenomena like wear and fatigue are rarely exact, however the range of variability can be narrowed. A component that lasts 1.1 service intervals is therefore more desirable than an expensive one that lasts for 1.9 service intervals. When designers talk about the same issues from a safety rather than a cost point of view, they often speak of redundancy in the products that enables the product to deal with unexpected situations. For example the user adds 2 additional kilos to the beam that can carry 7 kg, no problem will arise. However if they add 3 the beam might break prematurely. Although the product might not be used as intended, a failure could still affect the brand image or the safety of the product, so that companies plan in redundancy. These redundancies are, however, likely to be included in the functional requirements.

Margins are also added from a manufacturing and assembly point of view to ensure that a product will be safe and reliable for a given manufacturing variability. This is considered in robust design [39]. Tolerances can be considered as margins to accommodate manufacturing variability and are typically much smaller than margins in the sense of room for growth. However when margins become critical during a design process, tolerances might be affected by design margins.

Many designers discuss margins indirectly by referring to the ability of components to absorb change when incorporated into new designs. They lack the explicit concept of a margin, which is the subject of this research.

TEMPERATURE MARGINS IN JET ENGINE DESIGN

In the example of jet engine the overall performance of the engine, as an integrated system, (mainly propulsive thrust, but also power generation to the aircraft) is highly dependent on how well the constituent components are defined, optimized and matched together.

A gain in engine performance needs to be traded against the cost of reductions in structural integrity due to increase in heat stress. Higher gas temperatures are handled by either more advanced materials or by introducing cooling mechanisms or other smart design solutions to reduce the impact of high temperatures. The example (see Figure3) illustrates the coupled behavior between engine component and engine system, where both the magnitude and precision in gas temperatures – a key load parameter for designing hot structures– evolves.

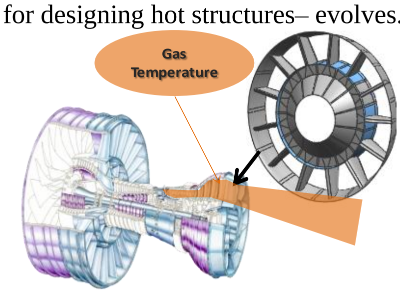

Figure 3 A rear turbine structure of a jet engine exposed to hot gases

Table 1 displays how the (fictitious) gas temperature evolves throughout the design process. Tg_1 is the design temperature at the time entering the development work. Tg_2 represents the update of the temperature following a system level trade of performance parameters. Tg_3 is the temperature as measured from the first physical test.

Table 1 Gas temperature evolution

Gas Temperature	Value (Degrees)	Comment
Tg_1	620	Initial value
Tg_2	660	First update
Tg_3	720	First measure

The inlet temperature to the turbine is a relevant example since – simplified – the higher the gas temperature at the turbine inlet, the higher the overall engine performance. The limitation on allowed temperature depends on how well the structural components – exposed to the hot gasses – can withstand the heat. The life length of the structural components is highly dependent on the material temperature, and the induced strains due to thermal variations induced.

Table 2 illustrates a simplified design problem by representing three candidate design solutions for the turbine structure, where the only design variable is the material, having different allowable design temperatures.

Table 2 Alternative design options

Design Solutions	Value (Degrees)	Comment
Design A	650	Design using Material A
Design B	710	Design using Material B
Design C	730	Design using Material C

The component manufacturer has the option to select between a well-known established material (Material A), an advanced material (Material B) and a novel material (Material C). Initially, the expected operative gas temperature T_{g1} is expected to be 620 degrees – indicating that the design can be made in material A with a margin of 30 degrees.

Two other aspects differentiate Material A, B and C. The cost of Material B is 70 % higher per unit weight that Material A. The cost of manufacturing processes (machining, welding etc) are also higher for Material B compared to material A. Material C is novel with no previously commercial application. The material cost is the same as for material B per unit weight, whereas the maturity of manufacturing and design is considerably lower. For temperatures greater than 700 the only alternative to using material B and C is to introduce some heat shielding/cooling technique. Such solutions increase weight and complexity of the design and are costly to introduce in later stages of the design work since the engine system is affected. The readiness of such design solutions or materials is low, which mean that there is a significant risk – and associated mitigation costs.

Managing the design process

Managing this process is a matter of managing expected and actual margins on components. The company has mature technology that can meet 620 Degrees (Tg_1) and less mature technology that can operate under a temperature of 660 (Tg_2). At the initial state of design, there was no discussion on the margins for future updates. Tg_1 was the best prediction of the final temperature. One strategy for the design team is to assume that the final temperature will not exceed 650 degrees and stay with their existing technology, knowing that if the temperature reaches a higher level, they will have to go through a design iteration to adapt to the TRL 6 technology. Alternatively they can assume at the beginning that there will be new design conditions, updated by the engine integrator, up to 710 degrees and develop their TLR technology. This would lead them to plan in a higher design afford at the beginning and depending on strategic decisions potentially offer the product at a high offer price. When the Tg_2 is updated to 660 the same logic applies; whether to trust this value and stick with the TLR 6 technology or look for a material that gives them a considerable higher margin. However, the latter option also introduces a much higher development risk. From a maturity point of view, the TRL level is a crucial "parameter" in that it represents a way to assess the reliability and maturity of a design option. Again, TRL levels no less than TRL 6 should be used in a design phase.

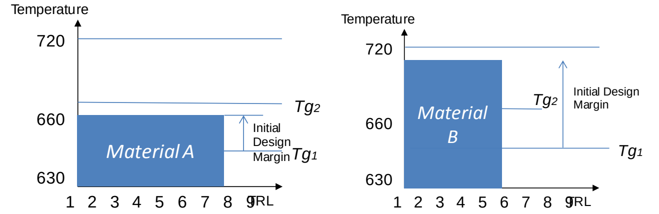

Figure 4 Technology readiness and temperature at the beginning of the process

Figure 4 illustrates the technology readiness levels for the two different technologies and the margins associated with it. Material A meets the current target, but has little margin, whereas material B has a large margin. Selecting Material B from the onset means accounting for extra cost and additional validation efforts that cannot be justified from the initial temperature assessments. Considering Material C to further ensure a design margin the company has to make changes to the other aspects of the design. Therefore, they have a round of internal iteration to accommodate changes. If the company was not able to bring material C to sufficient TRL they would have a number of options. They could bounce the change back to their customers declaring that they cannot accommodate the change. This risks the long collaboration with the customer. They could offer the solution using material B at their own risk and potentially be liable for the cost of engine failure or develop an alternative to material B. The company might take a long term view of increasing gas temperature and considering Material C, which only has TLR 4, but would give them a huge margin for the growth in temperature.

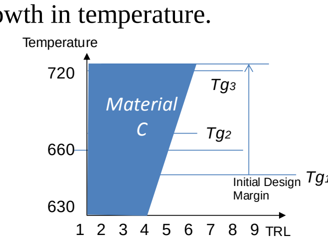

Figure 5 Increase of TLR over time

Figure 6 summarizes the design scenario, where the evolution of the design temperature is introduced during the development process, leaving the design team with several optional design strategies.

The uncertainty in predicting the design temperature can give rise to a design margin between the allowable design temperature and actual. Meeting the higher temperatures, such as the first measured temperature (Tg3), calls for advanced options. The introduction of Material C requires a maturation of the technology as well as accounting for a significantly higher cost. Due to the discrete "nature" of materials capability, the design solutions may be combined with heat reducing technologies (cooling). Such design solutions are not presented in the example and may be considered as alternatives to advanced and expensive materials. The drawback is that design options that require cooling technologies require an engine architecture that accounts for cooling of structures, which would need to be considered in advance of the initial designs.

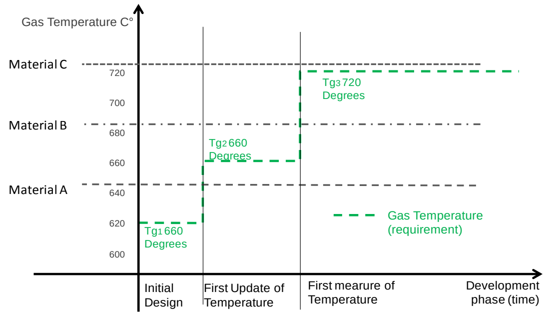

Figure 6 Temperature Increase vs Design Alternatives

DISCUSSION

The example illustrates how an understanding of design margins could help companies to have more effective product development processes and reduce the need for iterations. In problems as complex as jet engine design a certain degree of iteration is inevitable, as each player aims to optimize their own components and systems. However, unnecessary iteration occurs when one team does not know where another teams is aiming at and what they are likely to achieve.

Given the TRL levels in the example, the company will probably, in turn, continue to develop each of the materials to keep up with overall engine developments. Problems arise when they need to bring new materials on board, which are not developed to a sufficient TRL and need to be validated under unreasonable time pressures. Moving a new technology up several TRL requires time that can only be accelerated to a limited extent, because of the tests required to move through technology readiness levels. If a new technology can be brought on as a parallel development whilst deploying a mature technology, i.e. as a planned repetition and not as iteration in a process, then the company can plan its resources and have technology at a high TRL in place for the next generation of engines.

It would not be practical or sensible to trace all margins on all the components, but for each product there is a range of key parameters which determine the performance of the overall product, such as the engine temperature. A company will knows that these values will be changed and other values are expected to accommodate this. Another important group of margins to capture are those on critical components where a company knows that they will not be able to absorb change. If they are aware they might be able to compensate through making fundamentally different design choices. For example, instead of choosing a material that can withstand a higher temperature, the company could opt to cool their components. This might increase the design effort and potentially increase product complexity, but can be done with sufficient advance warning. This critical margin information can be flagged up and shared across an organization. This is happening already to some extent, but not in a formal and structured manner. Unexpected late iteration cycles can be a worst case scenario.

CONCLUSION

Iteration is an essential part of design processes. Companies want to control their iteration cycles, rather than being pushed into unexpected iteration through problems or miscommunication in design processes. Understanding and tracking margins on critical components and among components in subsystems, is an important way to better understand where and when iterations are required.

Future research will address the role of design margins in different design activities, and develop mathematical models to support the effective management of design margins.

ACKOWLEDGEMENT

The authors wish to acknowledge Volvo and GKN for funding the research contributing to his paper and Safaa Lebjioui participating in interviews on design margins.

REFERENCES

[1] Wynn, D.C., Eckert, C.M., and Clarkson, P.J., 2007, “Modelling iteration in engineering design”, ICED 07.

[2] Braha, D. and Bar-Yam. Y, 2007, “The Statistical Mechanics of Complex Product Development: Empirical and Analytical Results”. Management Science, 53 (7). pp. 1127-1145.

[3] Mankins, J. C., 1995, "Technology Readiness Levels: A White Paper". NASA, Office of Space Access and Technology,Advanced Concepts Office.

[4] Yassine A., Joglekar, N., Braha, D., Eppinger S., and Whitney, D., 2003, “Information Hiding in Product Development: The Design Churn Effect”. Research in Engineering Design, 14(3). 131-144. 2003.

[5] Isaksson, O., Eckert, C.M., and Lindroth, P., 2014, “Optimization of products verses optimization of product platforms: an engineering change margins perspective”, International Design Conference- Design 2014, May 19 – 22, Dubrovnik.

[6] Wynn, D.C. and Clarkson, P.J., 2005, “Models of Designing”, in: “Design Process Improvement: A review of current practice”, Clarkson, P.J. and Eckert, C.M. (Eds.), Springer.

[7] Suh, N. P., 1990, “The Principles of Design”, Oxford University Press.

[8] Safoutin, M.J., 2003, “A Methodology for Empirical Measurement of Iteration in Engineering Design Processes”, PhD thesis, University of Washington.

[9] Jun, H.-B., and Suh, H.-W., 2008, “A modeling framework for product development process considering its characteristics”, IEEE Transactions of Engineering Management., 55(1), pp.103–119.

[10] Chusilp, P. and Jin, Y., 2006, “Impact of mental iteration on concept generation”, Journal of Mechanical Design, 128, pp. 14-25.
[11] Eppinger, S.D., 1991, “Model-based approaches to managing concurrent engineering”, Journal of Engineering Design, 2(4), pp. 283–290.
[12] Cho, S.H., and Eppinger, S.D., 2005. ”A Simulation-based Process Model for Managing Complex Design Projects.” IEEE Transactions on Engineering Management, 52(3).
[13] Smith, R.P., and Eppinger, S.D., 1997, ”Identifying Controlling Features of Engineering Design Iteration.” Management Science, 43(3), pp. 437-451.
[14] Krishnan, V., Eppinger, S. D., and Whitney, D. E., 1997, "A Model-Based Framework to Overlap Product Development Activities," Management Science, pp. 437-451.
[15] Yassine, A. and Braha, D. (2003) “Complex Concurrent Engineering and the Design Structure Matrix”, Approach. Concurrent Engineering: Research and Applications. Vol. 11 (3). 165-177.
[16] Eger, T., Eckert, C.M., and Clarkson, P.J., 2005, “The role of design freeze in product development”. ICED05.
[17] Jarrett, T., Eckert, C.M., Caldwell, N., and Clarkson, P.J., 2011, “Engineering change: an overview and perspective on the literature”, Research in Engineering Design, 22(2) pp. 103-124.
[18] Eckert, C.M., Clarkson, P.J., and Zanker, W., 2004,”Change and Customisation in Complex Engineering Domains”, Research in Engineering Design, 15 (1), pp. 1-21.
[19] Ariyo, O.O., Eckert, C.M., and Clarkson, P.J., 2006 “Unpleasant surprises in the design of complex products: why do changes propagate?”, IDETC/CIE2006.
[20] Giffin, M., de Weck, O., Bounova, G., Keller, R., Eckert, C.M., and Clarkson, P.J, 2009, “Change propagation analysis in complex technical systems”, Journal of Mechanical Design, 131 (8).
[21] Pasqual M., de Weck O., 2011, “Multilayer Network Model for Analysis and Management of Change Propagation”, ICED’11.
[22] Eckert C.M., Clarkson, P.J., and Earl, C.F., 2005, “Predictability of change in engineering: a complexity view”,, IDETC/CIE2005..
[23] Keller, R., Eckert, C.M. and Clarkson, P.J., 2008, “Using an engineering change methodology to support conceptual design”, Journal of Engineering Design.
[24] Clarkson, P.J., Simons, C., and Eckert, C.M., 2004, "Predicting Change Propagation in Complex Design," Journal of Mechanical Design, 126(5), pp. 788-787.
[25] Cohen, T., Navathe, S., and Fulton, R., 2000,, ”C-FAR: Change Favorable Representation” Computer Aided Design, 32, 321–338.
[26] Grantham-Lough K., Stone MC, Tumer I., 2006, Prescribing and implementing the risk in early design (RED) method. In ASME 2006 Design Engineering Technical Conferences CDROM, Philadelphia, Pennsylvania, USA.
[27] Kocar V., and Akgunduz A., 2010, “ADVICE: a virtual environment for engineering change management”, Computers in Industry, 61(1), pp.15-28.
[28] Ahmad, N., Wynn, D.C., and Clarkson, P.J., 2013, “Change impact on a product and its redesign process: A tool for knowledge capture and reuse”, Research in Engineering Design, 24(3), pp. 219-244.
[29] Hamraz, B., Caldwell, N.H.M., and Clarkson, P.J., 2012, “A Multi-domain Engineering Change Propagation Model to Support Uncertainty Reduction and Risk Management in Design”. Journal of Mechanical Design, 134 (10)
[30] Bucciarelli, L.L., 1994, “Designing Engineers”, Cambridge, MA, MIT Press.
[31] Kleinsmann, M., and Valkenburg, R., 2008, “Barriers and enablers for creating shared understanding in co-design projects”, Design Studies, 29, pp. 369-386.
[32] Star, S., 2010. “This is not a boundary object: Reflections on the origin of a concept”, Science, Technology & Human Values, 35(5), pp. 601–617.
[33] Crook, T.R., and Combs, J.G., 2007. “Sources and consequences of bargaining power in supply chains”, Journal of Operations Management, 25, pp. 546-555.
[34] Stacey, M.K., and Eckert, C.M., 2003, “Against Ambiguity”, *Computer Supported Cooperative Work*, 12 (2), pp. 153-183.
[35] Eckert, C.M., Earl, C.F., and Isakson, O., 2012, “Product Property Margins: An underlying critical problem of engineering design”, *TMCE 2012*, May 7–11, 2012, Karlsruhe, Germany, Edited by I. Horváth, A. Albers, M. Behrendt and Z. Rusák.
[36] Papalambros, P., 2000, “Principles of Optimal Design: Modeling and Computation”, Cambridge University Press, New York.
[37] Ullman, D. G., 1997, "The mechanical design process”, McGraw-Hill, New York
[38] Tahera, K., Earl, C. and Eckert, C., 2014, “Integrating virtual and physical testing to accelerate the engineering product development process”, Int. J. Information Technology and Management, 13 (2/3), pp.154–175
[39] Chen W., Allen, J. K., Tsui, K. L., and Mistree, F., 1996, “A Procedure for Robust Design: Minimizing Variations Caused by Noise Factors and Control Factors”, Journal of Mechanical Design, Transactions of the ASME,118.

