

# design system integration for aero engine design

Aero engine design is a complex process involving multiple disciplines, which collaborate iteratively to achieve an optimized design. This involves exchange of data (geometry, process files, CAE data and engineering significant parameters) at component, subsystem and system levels. The building blocks of this process have usually been in the form of in-house developed software.



**Tarun Kant**

Technology Solutions Leader - Engineering Software  
QuEST-Global Engineering  
[tarun.kant@quest-global.com](mailto:tarun.kant@quest-global.com)

# contents

1.0	Introduction	01
2.0	Design System Elements	02
3.0	Steps to decide the choice of an integration platform	03
4.0	Steps in Design System Integration	03
5.0	Conclusion	05
6.0	Author Profile	05

## Introduction

**1.0)** Aero engine design is a complex process involving multiple disciplines, which collaborate iteratively to achieve an optimized design. This involves exchange of data (geometry, process files, CAE data and engineering significant parameters) at component, subsystem and system levels. The building blocks of this process have usually been in the form of in-house developed software. Traditionally, these programs were developed

independently without adhering to common software standards and highly customized to work for specific problems. Further, these programs have their own input and output standards leading to creation of bespoke scripts to sequence them into a process. Finally, source or user documentation for such programs may not be available. Following picture demonstrates the aero engine design process and challenges faced.

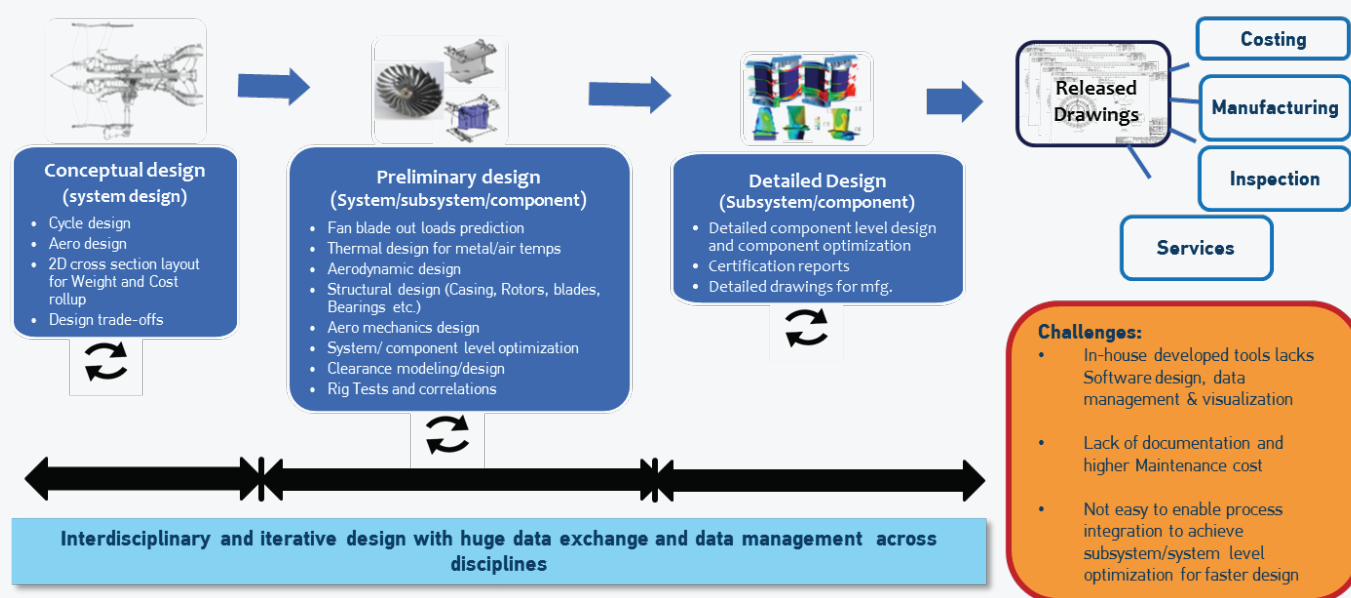


Figure - 1

Also the aero engine design is challenged to meet stringent targets on the following:

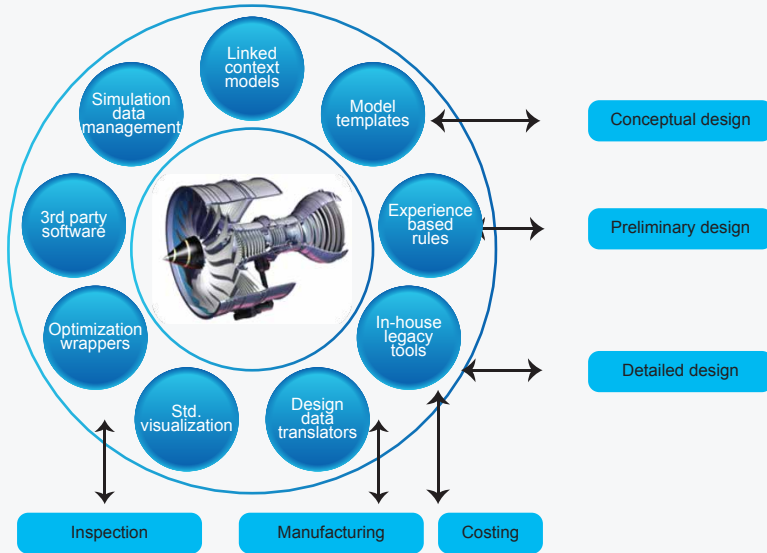
- Low specific fuel consumption
- Low emissions
- Low maintainability
- Low weight
- Low noise

These criteria impose additional challenge to aero engine design to reduce design margins and thereby improve their design methods. Also it requires multi-disciplines to collaborate more to achieve optimized and robust design.

All these factors demand a design system wherein design engineers can seamlessly perform system level, sub system level and multi-disciplinary design optimization and don't need to worry about model building, boundary conditions mapping, transferring of files from one program to another, etc.,. Therefore there is a strong need to build an integrated design system where multiple process steps from conceptual design to preliminary design to detailed design phase are seamlessly connected in a workflow and process steps and data are managed by the design system.

Also most of the engineering analysis (CFD, FEM, FBO etc.) techniques are highly computer intensive and usually require high performance computing. Lot of time is spent in moving the data from design engineer's computer to a High Performance Computing (HPC) set up and back, thereby taking more time and adding onto infrastructure cost. As the VDI (Virtual Desk-top Imaging) technology has improved the speed of visualization of

design results, it is becoming more feasible to have design systems run on HPC clusters. Visualization of results can be done in desktop using VDI thereby reducing the time of data transfer, better file/Simulation data management in HPC machines and most importantly reducing the cost of high end engineering machines used by each engineer. Following figure demonstrate the ideal state for a design system:



**Advantages:**

- Integrated design system capable of performing component, subsystem and system level optimization
- Data management
- Visualization at all level and allows users to have interactive design capability.
- Easy to plugin 3rd party software
- Easy to maintain
- Can run on HPC env. and use of VDI(Virtual desk top imaging)

Figure - 2

## Design System Elements

2.0) The design system essentially should have an architecture that integrates process steps that calls standard methods, translates data from one step to another, incorporates a common visualization engine for engineering significant parameters and most importantly works on common geometry models. All of this should be backed by good data management system (geometry data and simulation data). From software standpoint the

development environment should be using best of software engineering principles such as requirement specification documents, software design, configuration management, integrated development environment, test standardization and automation, release management and customer support management. Following figure explains the essential elements of design system:

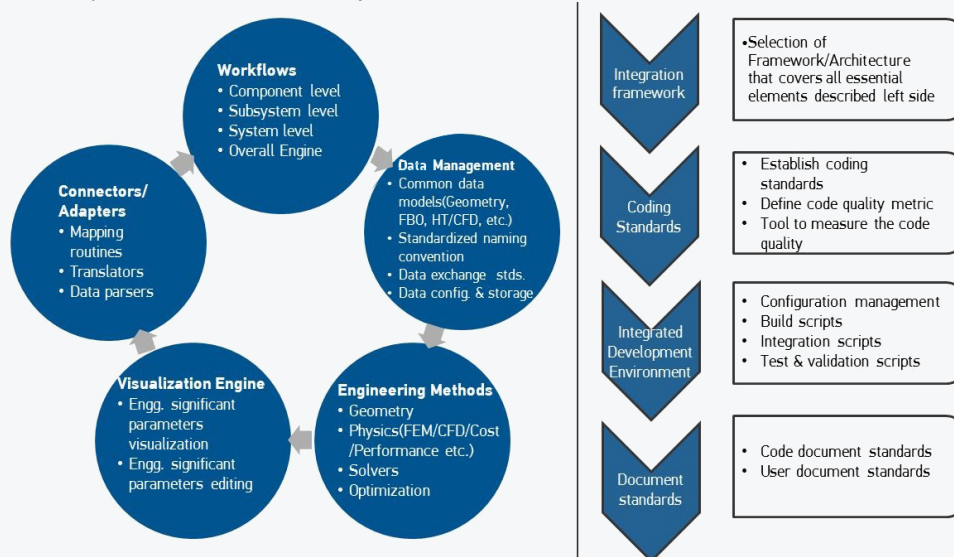


Figure - 3



## Steps to Decide the Choice of an Integration Platform

**3.0)** There are number of commercially available integrated design system packages like Isight, Model centre, Mode frontier and AML (Adaptive modelling language). Each one of them has unique advantages over other. Some systems are strong in optimization methods while some are strong in interfacing with commercial software and in-house processes. Some are easy to program and can help create complex work flow. The choice really depends upon the organization's current design system, processes used within each department and design needs. Also one needs to see what all processes will be part of the design system and what all will stay out. This decision depends on the ROI/complexity of the design process.

Looking at overall design needs one needs to study the design process in detail, look out for current tools (in-house and COTS) and come up with criterion. Then each of the system needs to be scored against the criterion with some weights and arrive at final recommendation.

Major factors that decide particular software should include following factors:

- Operating system support
- Analysis integration
- Analysis distribution
- Optimization methods
- Visualization
- Data collection
- Ease of Use
- Cost

Care should be taken to write functionality and process steps in most generic way so that that it is easy to move from one platform to other.

## Steps In Design System Integration

**4.0)** Design system integration requires the following steps:

- Decide and define common integration platform to integrate component, subsystem & system level design requirements
- Build standard and common geometry models used by different disciplines and at different stages of design cycle
- Define data standardization involving engineering significant parameters, naming conventions standardization
- Define integration plugins to Product data management systems and COTS software
- Define Visualisation engine that is used across all the work streams
- Define gaps in the current methods in reference to design system requirements
- Conversion of existing / legacy codes as per new architecture and coding standard
- Upgrade / create methods per automated design system needs (methods + software development)
- Map design practices / best practices into standard work flow at component, subsystem and system levels
- Identify gaps in the process and automate any gaps
- Create data interface adapters between process steps and between processes
- Choosing the right optimization methods as per the design requirements and ensuring that integration platform has the required optimization methods
- Automating the work flows and integration of workflows from component to system level
- Building test and validation cases (interdisciplinary model building) and perform automated testing (unit, functional and regression)
- Package and deploy these applications (production servers, HPC environment) ensuring configuration management

- Sunset plan of existing production systems
- Documentation (release notes, developer docs, user docs)
- User support (training, bug fixes and incident management)

We believe that to execute the above work scope, in addition to the domain and software skills, it also

requires strong Software Development Life Cycle (SDLC) knowledge. While our teams support all the three popular SDLC models (Waterfall Model, V Model and Agile Method), for building design system it is highly recommended to follow the Agile Method.

## Agile Application Development Process

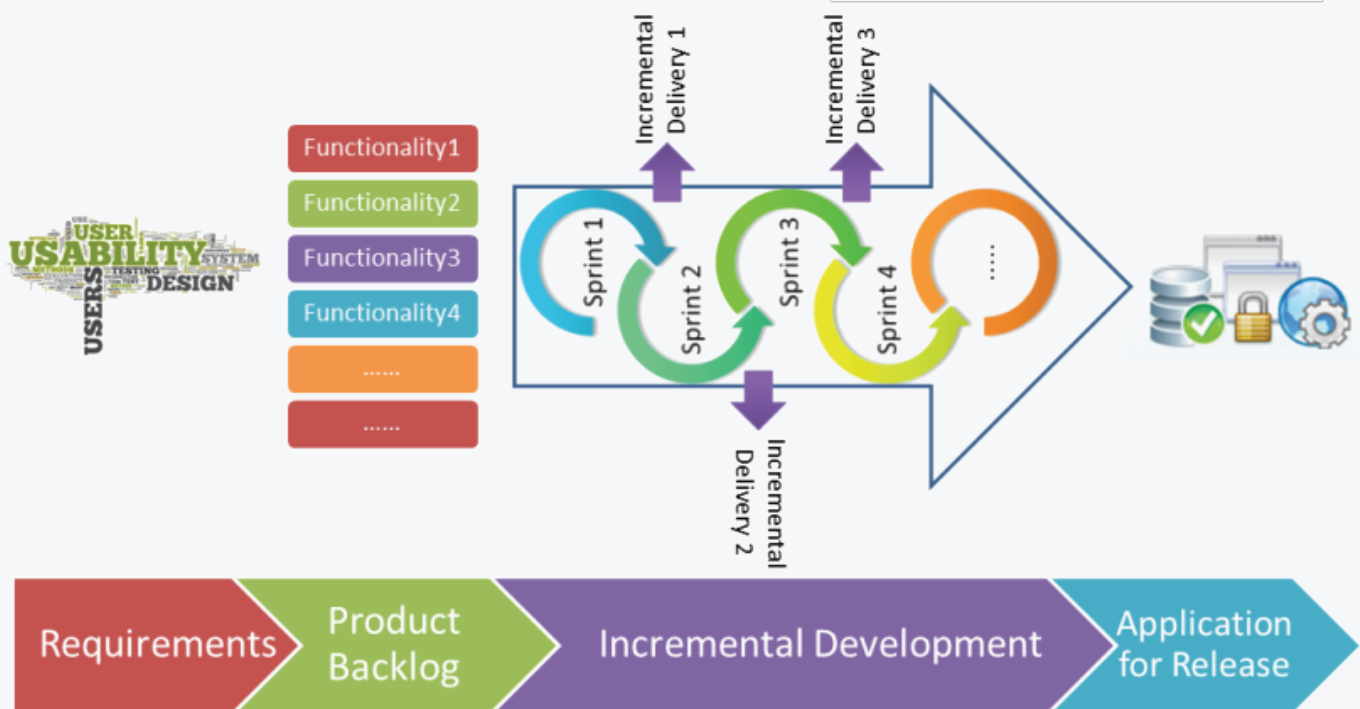


Figure - 4

Through the Agile Application Development Process, the user will get clear visibility of application development and can start using the application incrementally. It also

gives the flexibility to revise requirements periodically based on the evolving needs of application as mapped to individual design processes and integration.

### i. Requirement Analysis

- Identify customer needs and prioritize system requirements
- Establish schedule and constraints
- Identify product owner, scrum master and roles & responsibilities of team members

### ii. Product Backlog

- Translate system requirements to software / application requirements
- Identify and prioritize functionalities required for the application
- Prepare product backlog, user stories

### iii. Incremental Development

- Identify latest top priority product backlog item and prepare sprint backlog
- Conduct daily scrum meeting and facilitate to address road blocks, if any
- Refine and update product backlog from time to time in coordination with product owner
- Plan and release latest functionality to end users (Build, Test and Release)

### iv. Application Release

- Deploy application to targeted users
- Collect feedback and incorporate necessary changes / updates
- Full production release of the application
- Maintenance and support



## Conclusion

**5.0)** Design system integration is essential in achieving state-of-the-art aero engine design and is the key to enable faster and robust aero engine design. This also helps in driving down cost of maintenance of hardware and software. Commercial design system integration software provide platform to integrate design but lot of

effort is needed in defining the workflow process, migrating/modernizing legacy methods, creating common geometry models and automating processes using in-house / commercial tools. This requires engineering, software and process teams to work together and draw best from each other.

## Author Profile



### Tarun Kant

Tarun leads Engineering Software practice at QuEST. In this role Tarun works with QuEST's customers to provide solutions to address design, analysis and manufacturing software automation solutions, engineering analytics solutions and PLM customization. He leads pre-engagement activities which include working with functional leaders at our customers to map their engineering processes and help determine right automation solution. He is currently based out of Bangalore.

Tarun is a GE veteran with overall 13+ years of engineering software experience. He has strong design, analysis automation expertise in aircraft engines, gas/steam turbines, etc. He has worked in multiple CAD, CAE and software integration platforms and has strong domain expertise. At GE, he was leading multiple teams for automating fan, compressor and turbine design systems that includes areas like computational geometry, reverse engineering, manufacturing tools, CAD, aero mechanics, stress and life of rotating components composites, impact Analysis and aero dynamics tools development. In his last role at GE, he led the team on designing wing structure components for airbus next generation aircraft. He has strong experience in building new methodology for improving fidelity of design/analysis. During his career at GE, he has led global projects and worked with various GE offices in US, UK, Poland and India.

Tarun holds Bachelor in Mechanical Engineering from Punjab Engineering College Chandigarh, and Masters in Technology from IIT Delhi. He has also been through various GE management leadership programs during his GE tenure.



## About QuEST Global

---

QuEST Global is a focused global engineering solutions provider with a proven track record of over 17 years serving the product development & production engineering needs of high technology companies. A pioneer in global engineering services, QuEST is a trusted, strategic and long term partner for many Fortune 500 companies in the Aero Engines, Aerospace & Defence, Transportation, Oil & Gas, Power, Healthcare and other high tech industries. The company offers mechanical, electrical, electronics, embedded, engineering software, engineering analytics, manufacturing engineering and supply chain transformative solutions across the complete engineering lifecycle.

QuEST partners with customers to continuously create value through customer-centric culture, continuous improvement mind-set, as well as domain specific engineering capability. Through its local-global model, QuEST provides maximum value engineering interactions locally, along with high quality deliveries at optimal cost from global locations. The company comprises of more than 7,000 passionate engineers of nine different nationalities intent on making a positive impact to the business of world class customers, transforming the way they do engineering.



BORN TO ENGINEER

<http://quest-global.com>