QUICK REFERENCE GUIDE
LEAN ENABLERS FOR SYSTEMS ENGINEERING (LEfSE)

The LEfSE were developed by the Lean Systems Engineering Working Group of INCOSE (International Council for Systems Engineering). Version 1.0 was released at the INCOSE IW, San Francisco, March 1, 2009.

This brochure is intended to act as a day-to-day working reference for managers and practitioners involved in systems engineering and complex product development, to help them spot improvement opportunities and identify and apply the relevant Lean Enablers to improve their own and their organization's performance. It lists many of the forms of waste typically observed in systems engineering and product development efforts, and then provides a comprehensive list of the Lean Enablers, presented in tabular form.

LEfSE is a collection of 194 practices and recommendations formulated as "dos" and "don'ts" of SE based on Lean Thinking. The practices cover a large spectrum of SE and other relevant enterprise management practices, with a general focus to improve program value and stakeholder satisfaction, and reduce waste, delays, cost overruns, and frustrations.

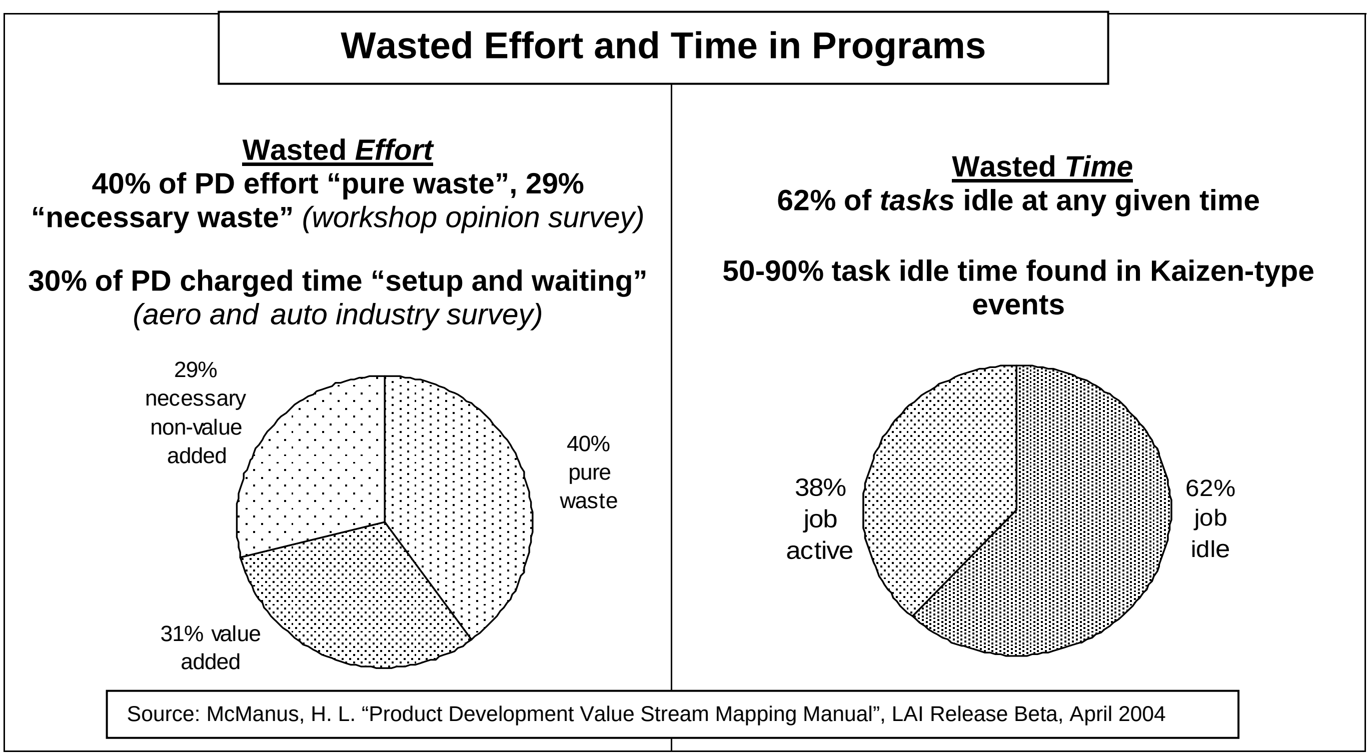
LEfSE are grouped into the six Lean Principles. The LEfSE are not intended to become a mandatory tool. Instead, they should be used as a checklist of good holistic practices.

Some enablers are intended for top enterprise managers, some for programs, and others for line employees. Some are more actionable than others, and some are easier to implement than others. Some enablers may require changes in company policies and culture. However, employee awareness of even those least actionable and most difficult to implement enablers should improve the thinking at work.

Recommended implementation: training and reading.

The full text of the LEfSE is available online: http://cse.lmu.edu/about/graduateeducation/systemsengineering/INCOSE.htm

The Opportunity: 40-70% potential cost and time improvement by removing waste



The problem – waste in systems engineering may be difficult to recognize

Seven categories of Waste adopted from Millard 2001:

WASTE	DESCRIPTION / EXAMPLES
1. Overproduction	• Creating unnecessary information, too much detail • Performing work which is not needed • Creating documents that nobody requested • Over dissemination • Sending a volume when a single number was requested • Reinventing the wheel • Reinventing discarded knowledge (layoffs) • Measuring mediocrity instead of pursuing perfection
2. Transportation	• Inefficient transmittal of information • Communication failure: lost data, information incompatibility, misunderstandings • Transportation for approvals • Multiple sources or destinations • Security issues • Distributed facilities
3. Waiting	• Waiting for data, test result, information, decision, signature,... • Poor planning of task precedence • Poor coordination • Disorganization, reorganization • Unnecessary serial effort
4. Over Processing	• Working more than necessary to produce the outcome • Point design used too early, causing later massive iterations • Starting with too small margins causing massive iterations • Uncontrolled iterations (too many tasks iterated) • Work on a wrong release (information churning) • Data conversions • Some contractual obligations (e.g., 2D drawings) • Unclear requirements or too many detailed requirements • Complex software monuments (using complex s/w when a spreadsheet would do)
5. Inventory	• Keeping more information than needed • Poor configuration management • Multiple databases and complicated retrieval • Poor 5S's in factory or office
6. Unnecessary movement	• People having to move to access information • Manual intervention to compensate for the lack of process • Hand-offs
7. Defects	• Insufficient quality of information • The "killer R's": Rework, Reprogram, Recalibrate, Recertify, Recheck, Resend, Retest, Re-inspect, Re-measure, Rewrite, etc. • Incomplete, ambiguous or inaccurate information

Note: Experience indicates that waste in engineering programs is difficult to recognize. The wastes of waiting and defects are the least controversial because engineers do not feel threatened by attempts to eliminate rework by perfecting the process, or to eliminate waiting for data by better preparations, planning, coordination and communication. The remaining categories of waste tend to be less obvious and more controversial. For example, the definition of Over-processing: "Working more than necessary to produce the outcome" is sometimes met with retort "refinement is always good". A person skilled in Lean Thinking would dispute with: "refinements are good only up to the points of satisfying the requirements and the need for *first time right*, and become waste beyond that."

The solution: The Lean Enablers for Systems Engineering

LEfSE are grouped into the six Lean Principles:

- Under the **Value Principle**, the enablers promote a robust process of establishing the value of the end-product or system to the customer with crystal clarity. The process should be customer-focused, and aligning the enterprise employees accordingly.
- The enablers under the **Value Stream Principle** emphasize waste-preventing measures, solid preparation of the personnel and processes for subsequent efficient workflow; detailed program planning; frontloading; and use of leading indicators and quality metrics.
- The **Flow Principle** lists the enablers which promote the uninterrupted flow of robust quality work and first-time right; steady competence instead of hero behavior in crises; excellent communication and coordination; concurrency; frequent clarification of the requirements; and making program progress visible to all.
- The enablers listed under the **Pull Principle** are a powerful guard against the waste of rework and overproduction. They promote pulling tasks and outputs based on need and rejecting others as waste.
- The **Perfection Principle** promotes excellence in the SE and enterprise processes; the use of the wealth of lessons learned from previous programs in the current program; the development of perfect collaboration policy across people and processes; and driving out waste through standardization and continuous improvement. A category of these enablers calls for a more important role of systems engineers, with responsibility, accountability and authority for the overall technical success of the program.
- Finally, the **Respect-for-People Principle** contains enablers that promote the enterprise culture of trust, openness, respect, empowerment, cooperation, teamwork, synergy, good communication and coordination; and enable people for excellence.

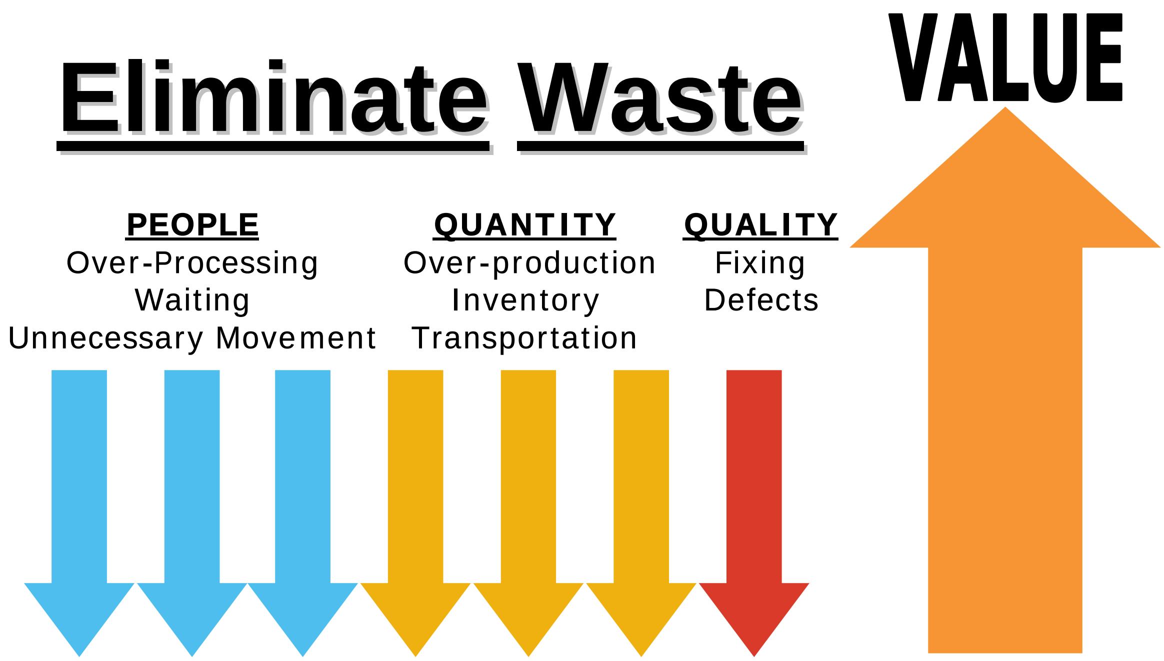
The full set of enablers is listed below in tabular form with one box for each Lean principle.

Enablers for Lean Principle 1: Value
Summary: The enablers promote a robust process of establishing the value of the end-product or system to the customer with crystal clarity, and frequently involving the customer.
1. 1 Follow all practices for the requirements capture and development in the INCOSE Handbook. In addition:
1.2 Establish the Value of the End Product or System to the Customer
1.2.1 Define value as the outcome of an activity that satisfies at least three conditions,
1.2.1.a. The external customer is willing to pay for "Value"
1.2.1.b. Transforms information or material or reduces uncertainty
1.2.1.c. Provides specified performance right the first time
1.2.2 Define value-added in terms of value to the customer and his needs
1.2.3 Develop a robust process to capture, develop, and disseminate customer value with extreme clarity
1.2.4 Develop an agile process to anticipate, accommodate and communicate changing customer requirements
1.2.5 Do not ignore potential conflicts with other stakeholder values, and seek consensus
1.2.6 Explain customer culture to Program employees, i.e. the value system, approach, attitude, expectations, and issues
1.3 Frequently Involve the Customer
1.3.1 Everyone involved in the program must have a customer-first spirit
1.3.2 Establish frequent and effective interaction with internal and external customers
1.3.3 Pursue an architecture that captures customer requirements clearly and can be adaptive to changes
1.3.4 Establish a plan that delineates the artifacts and interactions that provide the best means for drawing out customer requirements.

Enablers for Lean Principle 2: Map the Value Stream (Plan the Program)
Summary: These enablers emphasize waste-preventing measures, solid preparation of the personnel and processes for subsequent efficient workflow and healthy relationships between stakeholders (customer, contractor, suppliers, and employees); detailed program planning; frontloading; and use of leading indicators and quality metrics.
2.1 Plan the Program according to the INCOSE Handbook Process. In addition:
2.2 Map the SE and PD Value Streams and Eliminate Non-Value Added Elements
2.2.1 Develop and execute clear communication plan that covers entire value stream and stakeholders
2.2.2 Have cross functional stakeholders work together to build the agreed value stream
2.2.3 Create a plan where both Systems Engineering and other Product Development activities are appropriately integrated.
2.2.4 Maximize co-location opportunities for SE and PD planning (SE is a part of PD. In this paragraph, the PD should be understood as denoting all PD activities other than SE, including design, development, manufacturing, integration, testing, etc.)
2.2.5 Use formal value stream mapping methods to identify and eliminate SE and PD waste, and to tailor and scale tasks
2.2.6 Scrutinize every step to ensure it adds value, and plan nothing because "it has always been done"
2.2.7 Carefully plan for precedence of both SE and PD tasks (which task to feed what other task(s) with what data and when), understanding task dependencies and parent-child relationships
2.2.8 Maximize concurrency of SE and other PD Tasks
2.2.9 Synchronize work flow activities using scheduling across functions, and even more detailed scheduling within functions
2.2.10 For every action, define who is responsible, approving, supporting, and informing ("RASI"), using a standard and effective tool, paying attention to precedence of tasks
2.2.11 Plan for level workflow and with precision to enable schedule adherence and drive out arrival time variation.
2.2.12 Plan below full capacity to enable flow of work without accumulation of variability, and permit scheduling flexibility in work loading, i.e., have appropriate contingencies and schedule buffers. (Queuing theory proves that the flow approaching 100% of capacity slows down asymptotically due to the accumulation of variability, even in the absence of any bottlenecks.)
2.2.13 Plan to use visual methods wherever possible to communicate schedules, workloads, changes to customer requirements, etc.
2.3 Plan for Frontloading the Program
2.3.1 Plan to utilize cross-functional teams made up of the most experienced and compatible people at the start of the project to look at a broad range of solution sets
2.3.2 Explore trade space and margins fully before focusing on a point design and too small margins.
2.3.3 Anticipate and plan to resolve as many downstream issues and risks as early as possible to prevent downstream problems.
2.3.4 Plan early for consistent robustness and "first time right" under "normal" circumstances instead of hero-behavior in later "crisis" situations.
2.4 Plan to Develop Only What Needs Developing
2.4.1 Promote reuse and sharing of program assets: Utilize platforms, standards, busses, and modules of knowledge, hardware and software
2.4.2 Insist that a module proposed for use is robust before using it
2.4.3 Remove show-stopping research/unproven technology from critical path, staff with experts, and include it in the Risk Mitigation Plan
2.4.4 Defer unproven technology to future technology development efforts, or future systems
2.4.5 Maximize opportunities for future upgrades, (e.g., reserve some volume, mass, electric power, computer power, and connector pins), even if the contract calls for only one item.
2.5 Plan to Prevent Potential Conflicts with Suppliers
2.5.1 Select suppliers who are technically and culturally compatible
2.5.2 Strive to develop seamless partnership between suppliers and the product development team
2.5.3 Plan to include and manage the major suppliers as a part of your team
2.5.4 Have the suppliers brief the design team on current and future capabilities during conceptual formation of the project.
2.6 Plan Leading Indicators and Metrics to Manage the Program
2.6.1 Use leading indicators to enable action before waste occurs
2.6.2 Focus metrics around customer value, not profits
2.6.3 Use only few simple and easy to understand metrics and share them frequently throughout the enterprise
2.6.4 Use metrics structured to motivate the right behavior
2.6.5 Use only those metrics that meet a stated need or objective.

Enablers for Lean Principle 3: Flow
Summary: These enablers promote the uninterrupted flow of robust quality work and first-time right; frontloading, steady competence instead of hero behavior in crises; excellent communication and coordination; concurrency; frequent clarification of the requirements; and making program progress visible to all.
3.1 Execute the Program according to the INCOSE Handbook Process. In addition:
3.2 Clarify, Derive, Prioritize Requirements Early and Often During Execution
3.2.1 Since formal written requirements are rarely enough, allow for follow up verbal clarification of context and need, without allowing requirements creep
3.2.2 Create effective channels for clarification of requirements (possibly involve customer participation in development IPTs)
3.2.3 Listen for and capture unspoken customer requirements
3.2.4 Use architectural methods and modeling for system representations (3D integrated CAE toolset, mockups, prototypes, models, simulations, and software design tools) that allow interactions with customers as the best means of drawing out customer requirements
3.2.5 "Fail early fail often" through rapid learning techniques (prototyping, tests, digital preassembly, spiral development, models, and simulation)
3.2.6 Identify a small number of goals and objectives that articulate what the program is set up to do, how it will do it, and what the success criteria will be to align stakeholders and repeat these goals and objectives consistently and often.
3.3 Front Load Architectural Design and Implementation
3.3.1 Explore multiple concepts, architectures and designs early
3.3.2 Explore constraints and perform real trades before converging on a point design
3.3.3 Use a clear architectural description of the agreed solution to plan a coherent program, engineering and commercial structures
3.3.4 All other things being equal, select the simplest solution ("Any fool can make anything complex but it takes a genius and courage to create a simple solution" - Albert Einstein.)
3.3.5 Invite suppliers to make a serious contribution to SE, design and development as program trusted partners
3.4 Systems Engineers to accept Responsibility for coordination of PD Activities
3.4.1 Promote maximum seamless teaming of SE and other PD engineers
3.4.2 SE to regard all other engineers as their partners and internal customers, and vice-versa
3.4.3 Maintain team continuity between phases to maximize experiential learning
3.4.4 Plan for maximum continuity of Systems Engineering staff during the Program
3.5 Use Efficient and Effective Communication and Coordination
3.5.1 Capture and absorb lessons learned from almost all programs: "never enough coordination and communication"
3.5.2 Maximize coordination of effort and flow (one of the main responsibilities of Lean SE)
3.5.3 Maintain counterparts with active working relationships throughout the enterprise to facilitate efficient communication and coordination among different parts of the enterprise, and with suppliers
3.5.4 Use frequent, timely, open and honest communication
3.5.5 Promote direct informal communications immediately as needed
3.5.6 Use concise one-page electronic forms (e.g., Toyota's A3 form) rather than verbose unstructured memos to communicate, and keep detailed working data as backup
3.5.7 Report cross-functional issues to be resolved on concise standard one-page forms to Chief's office in real time for his/her prompt resolution
3.5.8 Communicate all expectations to suppliers with crystal clarity, including the context and need, and all procedures and expectations for acceptance tests, and ensure the requirements are stable
3.5.9 Trust engineers to communicate with suppliers' engineers directly for efficient clarification, within a framework of rules, (but watch for high risk items which must be handled at the top level)
3.6 Promote Smooth SE Flow
3.6.1 Use formal frequent comprehensive integrative events in addition to programmatic reviews
3.6.1.a Question everything with multiple "whys"
3.6.1.b Align process flow to decision flow
3.6.1.c Resolve all issues as they occur in frequent integrative events
3.6.1.d Discuss tradeoffs and options
3.6.2 Be willing to challenge the customer's assumptions on technical and meritocratic grounds, and to maximize program stability, relying on technical expertise
3.6.3 Minimize handoffs to avoid rework
3.6.4 Optimize human resources when allocating VA and RNVA tasks
3.6.4.a Use engineers to do VA engineering
3.6.4.b When engineers are not absolutely required, use non-engineers to do RNVA (administration, project management, costing, metrics, program, etc.)
3.6.5 Ensure the use of the same measurement standards and data base commonality
3.6.6 Ensure that both data deliverers and receivers understand the mutual needs and expectations
3.7 Make Program Progress Visible to All
3.7.1 Make work progress visible and easy to understand to all, including external customer
3.7.2 Utilize Visual Controls in public spaces for best visibility (avoid computer screens)
3.7.3 Develop a system making imperfections and delays visible to all
3.7.4 Use traffic light system (green, yellow, red) to report task status visually (good, warning, critical) and make certain problems are not concealed
3.8 Use Lean Tools
3.8.1 Use Lean tools to promote the flow of information and minimize handoffs: small batch size of information, small takt times, wide communication bandwidth, standardization, work cells, training.
3.8.2 Use minimum number of tools and make common wherever possible
3.8.3 Minimize the number of the software revision updates and centrally control the update releases to prevent information churning
3.8.4 Adapt the technology to fit the people and process
3.8.5 Avoid excessively complex "monument" tools

Enablers for Lean Principle 4: Pull
Summary: These enablers are a powerful guard against the waste of rework and overproduction. They promote pulling tasks and outputs based on need (and rejecting others as waste) and better coordination between the pairs of employees handling any transaction before their work begins (so that the result can be first-time right).
4.1 Tailor for a given program according to the INCOSE Handbook Process. In addition:
4.2 Pull Tasks and Outputs Based on Need, and Reject Others as Waste
4.2.1 Let information needs pull the necessary work activities.
4.2.2 Promote the culture in which engineers pull knowledge as they need it and limit the supply of information to only genuine users
4.2.3 Understand the Value Stream Flow
4.2.4 Train the team to recognize who the internal customer (Receiver) is for every task as well as the supplier (Giver) to each task use a SIPCO (supplier, inputs, process, outputs, customer) model to better understand the value stream
4.2.5 Stay connected to the internal customer during the task execution
4.2.6 Avoid rework by coordinating task requirements with internal customer for every non-routine task
4.2.7 Promote effective real time direct communication between each Giver and Receiver in the value flow
4.2.8 Develop Giver-Receiver relationships based on mutual trust and respect
4.2.9 When pulling work, use customer value to separate value added from waste



A summary article describing the fundamental concepts of Lean Thinking: Value, Waste and the Six Lean Principles, as well as the LEfSE project can be found in *Lean Enablers for Systems Engineering*, B. W. Oppenheim, CrossTalk, July/August 2009, available at: http://cse.lmu.edu/Assets/Colleges+Schools/CSE/Mechanical+Engr/Crosstalk+LEfSE.pdf

A comprehensive presentation describing the development of the Lean Enablers for Systems Engineering is available at: http://cse.lmu.edu/Assets/Colleges+Schools/CSE/Mechanical+Engr/Lean_Enablers_for_SE_Version_1_01_.pdf

A comprehensive article describing the history of Lean Systems Engineering, the fundamental concepts of Lean Thinking: Value, Waste and the Six Lean Principles, the development process of LEfSE, the full text of the enablers, the supporting validation surveys, benchmarking with other studies, and industrial examples can be found in: *Lean Enablers for Systems Engineering*, B. W. Oppenheim, E. Murman, D. Secor, Journal of Systems Engineering, (in review), 2009.

Enablers for Lean Principle 5: Perfection
Summary: These enablers promote excellence in the SE and enterprise processes; the use of the wealth of lessons learned from previous programs; the development of perfect collaboration policy across people and processes; and driving out waste through standardization and continuous improvement. A group of the enablers calls for a more important role of systems engineers, with responsibility, accountability and authority for the overall technical success of the program.
5.1 Pursue Continuous Improvement according to the INCOSE Handbook Process. In addition:
5.2 Strive for Excellence of SE Processes
5.2.1 Do not ignore the basics of Quality:
5.2.1.a Build in robust quality at each step of the process, and resolve and do not pass along problems.
5.2.1.b Strive for perfection in each process step without introducing waste
5.2.1.c Do not rely on final inspection; error proof wherever possible
5.2.1.d If final inspection is required by contract, perfect upstream processes pursuing 100% inspection pass rate
5.2.1.e Move final inspectors upstream to take the role of quality mentors
5.2.1.f Apply basic PDCA method (plan, do, check, act) to problem solving
5.2.1.g Adopt and promote a culture of stopping and permanently fixing a problem as soon as it becomes apparent.
5.2.2 Promote excellence under "normal" circumstances instead of hero-behavior in "crisis" situations
5.2.3 Use and communicate failures as opportunities for learning emphasizing process and not people problems
5.2.4 Treat any imperfection as opportunity for immediate improvement and lesson to be learned, and practice frequent reviews of lessons learned
5.2.5 Maintain a consistent disciplined approach to engineering
5.2.6 Promote the idea that the system should incorporate continuous improvement in the organizational culture, but also...
5.2.7 ...balance the need for excellence with avoidance of overproduction waste (pursue refinement to the point of assuring Value and "first time right", and prevent over-processing waste)
5.2.8 Use a balanced matrix/project organizational approach avoiding extremes: territorial functional organizations with isolated technical specialists, and all-powerful IPTs separated from functional expertise and standardization
5.3 Use Lessons Learned from Past Programs for Future Programs
5.3.1 Maximize opportunities to make each next program better then the last
5.3.2 Create mechanisms to capture, communicate, and apply experience-generated learning and checklists
5.3.3 Insist on workforce training of root cause and appropriate corrective action
5.3.4 Identify best practices through benchmarking and professional literature
5.3.5 Share metrics of supplier performance back to them so they can improve
5.4 Develop Perfect Communication, Coordination and Collaboration Policy across People and Processes
5.4.1 Develop a plan and train the entire program team in communications and coordination methods at the program beginning
5.4.2 Include communication competence among the desired skills during hiring
5.4.3 Promote good coordination and communications skills with training and mentoring
5.4.4 Publish instructions for email distributions and electronic communications.
5.4.5 Publish instructions for artifact content and data storage: central capture versus local storage, and for paper versus electronic, balancing between excessive bureaucracy and the need for traceability
5.4.6 Publish a directory of the entire program team and provide training to new hires on how to locate the needed nodes of knowledge
5.4.7 Ensure timely and efficient access to centralized data.
5.4.8 Develop an effective body of knowledge that is historical, searchable, shared by team, and knowledge management strategy to enable the sharing of data and information within the enterprise.
5.5 For Every Program Use a Chief Engineer Role[1] to Lead and Integrate the Program from Start to Finish
5.5.1 The Chief Engineer role to be Responsible, with Authority and Accountability for the program technical success
5.5.2 Have the Chief Engineer role lead both the product and people integration
5.5.3 Have the Chief Engineer role lead through personal influence, technical know how, and authority over product development decisions
5.5.4 Groom an exceptional Chief Engineer role with the skills to lead the development, the people, and assure program success
5.5.5 If Program Manager and Chief Engineer are two separate individuals (required by contract or organizational practice), co-locate both to enable constant close coordination.
5.6 Drive out Waste through Design Standardization, Process Standardization, and Skill-Set Standardization
5.6.1 Promote design standardization with engineering checklists, standard architecture, modularization, busses, and platforms
5.6.2. Promote process standardization in development, management, and manufacturing
5.6.3 Promote standardized skill sets with careful training and mentoring, rotations, strategic assignments, and assessments of competencies
5.7 Promote All Three Complementary Continuous Improvement Methods to Draw Best Energy and Creativity from All Employees
5.7.1 Utilize and reward bottom up suggestions for solving employee-level problems
5.7.2 Use quick response small Kaizen teams comprised of problem stakeholders for local problems and development of standards
5.7.3 Use the formal large Six Sigma teams for the problems which cannot be addressed by the bottom-up and Kaizen improvement systems, and do not let the Six Sigma program destroy those systems.

[1] A frequent practice in recent U.S. governmental programs is to have two program managers: the "Program Manager" responsible for the program business success, and "Chief Systems Engineer" responsible for Systems Engineering. Numerous functional engineers are responsible for various technical areas. In some programs this causes split responsibilities, authorities and accountabilities, often with imperfect results. In contrast, many U.S. and overseas commercial programs use only one person fully responsible for the entire program success (both technical and business). The person is called by various names, e.g. Chief Engineer (very successful Toyota model, [Morgan and Liker, 2006], Product Manager, Product Engineer, or similar. Early U.S. aerospace programs also used extremely successful single-person "Chief Engineer" role (e.g., early Jack Northrop, Howard Hughes, Kelly Johnson of the Skunk Works, early NASA space programs, and others). [Murman, 2008] discusses some more recent successful programs with a single top manager in the dual technical and business leadership role. Since this document for INCOSE, a Council for Systems Engineering rather than entire program management, the editors have addressed only the technical role of the Chief Engineer, saying nothing whether that person should also be the overall manager of the program, or share the management with a separate business manager person. However, nothing in this document should be taken as promoting the dual-head model. The dual-head model is not required under the U.S. government acquisition policies, and is not promoted in the [INCOSE SE Handbook, 2007]

Enablers for Lean Principle 6: Respect for People
Summary: These enablers promote the enterprise culture of trust, openness, respect, empowerment, cooperation, teamwork, synergy, good communication and coordination; and enable people for excellence.
6.1 Pursue People Management according to the INCOSE Handbook Process. In addition:
6.2 Build an Organization Based on Respect for People
6.2.1 Create a vision which draws and inspires the best people
6.2.2 Invest in people selection and development to promote enterprise and program excellence
6.2.3 Promote excellent human relations: trust, respect, empowerment, teamwork, stability, motivation, drive for excellence
6.2.4 Read applicant's resume carefully for both technical and non-technical skills, and do not utilize computer scanning for keywords
6.2.5 Promote direct human communication
6.2.6 Promote and honor technical meritocracy
6.2.7 Reward based upon team performance, and include teaming ability among the criteria for hiring and promotion
6.2.8 Use flow down of Responsibility, Authority and Accountability (RAA) to make decisions at lowest appropriate level
6.2.9 Eliminate fear and promote conflict resolution at the lowest level
6.2.10 Keep management decisions crystal clear but also promote and reward the bottom-up culture of continuous improvement and human creativity and entrepreneurship
6.2.11 Do not manage from cubicle; go to the spot and see for yourself
6.2.12 Within program policy and within their area of work, empower people to accept responsibility by promoting the motto “ask for forgiveness rather than ask for permission”
6.2.13 Build a culture of mutual support (there is no shame in asking for help)
6.2.14 Prefer physical team co-location to the virtual co-location
6.3 Expect and Support Engineers to Strive for Technical Excellence
6.3.1 Establish and support Communities of Practice
6.3.2 Invest in Workforce Development
6.3.3 Assure tailored lean training for all employees
6.3.4 Give leaders at all levels in-depth lean training
6.4 Nurture a Learning Environment
6.4.1 Perpetuate technical excellence through mentoring, training, continuing education, and other means
6.4.2 Promote and reward continuous learning through education and experiential learning
6.4.3 Provide knowledge experts as resources and for mentoring
6.4.4 Pursue the most powerful competitive weapon: the ability to learn rapidly and continuously improve.
6.4.5 Value people for the skills they contribute to the program with mutual respect and appreciation
6.4.6 Capture learning to stabilize the program when people change
6.4.7 Develop Standards paying attention to human factors, including reading and perception abilities
6.4.8 Immediately organize a quick training in any new standard
6.5 Treat People as Most Valued Assets, not as Commodities

Comments and suggestions for changes or improvements should be sent to:

In U.K.: Hillary Sillitto: hillary.sillitto@incose.org

In U.S.: Bo Oppenheim, boppenheim@lmu.edu