

Quotations from Chairman David

*A Little Red Book of Truths to Enlighten and Guide
on the Long March Toward the COTS Revolution*



This report was prepared for the

*SEI Joint Program Office
HQ ESC/DIB
5 Eglin Street
Hanscom AFB, MA 01731-2116*

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

(signature on file)

*Mario Moya, Maj, USAF
SEI Joint Program Office*

This work is sponsored by the U.S. Department of Defense.

Copyright 1998 by Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be addressed to the SEI Licensing Agent.

This work was created in the performance of Federal Government Contract Number F19628-95-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

This document is available through Asset Source for Software Engineering Technology (ASSET): 1350 Earl L. Core Road; PO Box 3305; Morgantown, West Virginia 26505 / Phone: (304) 284-9000 or toll-free in the U.S. 1-800-547-8306 / FAX: (304) 284-9001 World Wide Web: <http://www.asset.com/> e-mail: sei@asset.com

Copies of this document are available through the National Technical Information Service (NTIS). For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. Phone: (703) 487-4600.

This document is also available through the Defense Technical Information Center (DTIC). DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center / Attn: BRR / 8725 John J. Kingman Road / Suite 0944 / Ft. Belvoir, VA 22060-6218 / Phone: (703) 767-8274 or toll-free in the U.S.: 1-800 225-3842.

Foreword

This little paper, presented tongue-in-cheek, is the result of numerous frustrating experiences stemming from the current climate in the Defense community. That climate is one of change: the highest-level Defense acquisition policies now embrace the widespread use of commercial products, together with novel business methods and processes, and generally aim at moving Government acquisition practices toward accommodating the marketplace.

This means that for everyone, both in the contractor and in the Government camps, we are all now engaged in the mighty exercise of modifying, in a deep and fundamental manner, the way that Defense systems are designed and built. The frustration I noted above stems from my sense of two things. First, I perceive that many of us are paying phenomenal lip service to the new directives and mandates, to the things that must be accomplished to meet these changed circumstances. “We’ve all got to start doing business differently” we are saying, loud and clear. And second, most of us—upper-level policy managers to low-level analysts, civilian contractors and DoD colonels—are confidently doing just what we’ve always done, thinking that somehow the need for change applies to everyone except us, and waiting for the guy in the next office to mend his ways.

This problem is really as old as mankind. True change, true rebirth occurs from within, not by someone else's orders. So regardless of executive fiat, policy papers, or mandates, genuine change is done only at the personal level, and inculcating a new paradigm into tens of thousands of individuals is only accomplished through tens of thousands of personal commitments. So the "new way of doing business" will be painfully slow to trickle down (or up!) to each person at every level, and very inefficient at that. While we are waiting for everyone to turn over that new leaf and think differently, millions of dollars will be wasted, and the needs of dozens of Defense systems will not be met.

Numerous persons have commented on this problem on many occasions. Through papers, through "lessons learned," through case studies, and through other such mechanisms, the issue has been talked about and discussed so much that there is apparently little left to say. But none of these papers have accomplished their goals, at least not in the sense that their authors (myself included) have intended. It is still the case that, as with all resolutions about altered behavior, new habits are remarkably difficult to bring about. In short, the crying need is for everyone throughout a very large and diverse community to internalize some fairly weighty precepts; pontification is of little value here. We need, in effect, to re-educate ourselves, each of us individually, so that this new paradigm can flourish.

In musing on this recently, it dawned on me that what I was lamenting has an interesting resonance with an earlier point in the 20th century, when some nations (in a quite different political milieu) made extensive use of good, old-fashioned propaganda. So I thought back thirty years to one of history's masters of propaganda and "re-education," and, admittedly in search

of comic relief, I put together this “little red book.” (While it is based on a fairly well-known model, that model is probably best known to those of us who lived through the turbulent era of the 60s and 70s.)

But while the format is intended as amusing, the essential content is actually quite serious. My goal was to place some down-to-earth realities in bold relief by means of brief quasi-aphorisms, each of which is related to a single small, pertinent issue. I intended that each of these “quotations” could stand alone as a self-contained, fairly obvious statement. It is true that not all of these statements are absolutely true in all cases, nor each applicable to every occasion. But they are true enough in most cases. And while I have no illusion that this “little red book” will instantly bring about our own “great leap forward,” with thousands of chanting programmers marching through the streets (the Acquisitional Revolution? who would make up the Gang of Four?), I do have some hopes that by provoking a few smiles, one or two of the items found here might come in handy for one or another program manager or system developer.

The majority of the aphorisms pertain to the use of commercial software products (“COTS”) in complex defense systems. It is with using COTS, or rather with the unexpected implications that come from using COTS, that most of the problems arise for managers and developers alike when they try to comply with recent policy directives. A few of the other aphorisms are related to the emerging use of “integrated product teams” (IPTs) as the backbone of project execution. Still others are not really particular to commercial products or to any novel ways to do business, but simply are age-old verities that bear repeating one more time.

The “little red book” is divided into seven sections: general precepts about COTS products, requirements, evaluation and design, maintenance, business processes, testing and debugging, and IPTs. These divisions are no more than conveniences, and many of these aphoristic statements repeat, wandering back and forth around the same set of issues. I have not worried too much about the presence of a certain redundancy, since my goal was not to tell a logically connected story from start to finish, but rather to touch on a number of topics in a variety of ways, sometimes seconding the same point through different examples. I do believe, however, that in the aggregate, a coherent message can be drawn: in switching to the “new mode of doing business,” particularly to the explicit bias toward commercial products for Defense systems, it is foolhardy for anyone to believe that the rewards of the new paradigm can be gotten without a considerable expenditure of personal effort. This effort may not be measurable in dollars, but there can be no doubt that it must be spent. To summarize by means of an aphorism: “You get what you pay for.”

The hoped-for audience is just about anyone in the Defense community, since the frustrations that I have experienced occurred on all sides and levels of the Acquisition hierarchy. And to repeat the initial point: each individual in the community has to change his or her habits, sometimes in deep and profound ways. The experience that comes from an entire career may now be a doubtful, or even an untrue guide for the new Acquisition paradigm. This is a painful prospect to all of us. But it is the reality that we face. The quicker we face it, the better off we’ll all be. So now, on to the Long March. Let the Revolution begin.

David Carney
Software Engineering Institute
Carnegie Mellon University
July 1, 1998

東 市

Undying Truths about the Army, the Merchant, and the Cat

(Some General Precepts about COTS Software)

Our armies no longer need skilled horsemen. But driving a tank takes even greater skill than riding a horse.

The shift to commercial products means less need for development and coding experts, but greater need for integration experts and evaluation experts. Thus, making extensive use of COTS products implies not a lessening of expertise, but only a shift in the domain of the expertise that is needed.

*Look carefully at the merchant before you buy his goods.
Are his children dressed in rags?*

Using a COTS product implies a certain trust of its vendor. It is often useful, therefore, to learn as much as possible about the vendor—what are his other business obligations, what is his financial condition—before making the decision to use his product.

*The cat cannot live in the wren's nest,
nor can the wren catch a mouse.*

The COTS product does only what its builders intended it to do, not what you want it to do. It was not built to your needs or specifications, but to a set of needs, specifications, and commercial pressures known only to its vendor. Following a COTS-based approach means accepting that the way products work is governed by economic and market constraints, not by the needs of a specific DoD system.

*When the soldiers in the field get new weapons,
everyone in the camp feels different, even the cook.*

The impact of a shift to a COTS-based mode of acquisition is not restricted to developers and coders. Everyone, policy makers, test personnel, integrators, and managers alike will experience some change in the work they do and the way they do it.

The village merchant is not always a thief.

Seek the lowest price you can, but don't always presume that the vendor is venal or corrupt. Especially, don't begrudge the COTS vendor a just profit.

*The soldier should spend his time in fighting the enemy;
he should depend on the cat if he needs to kill a mouse*

While cost savings are indeed hoped for, the impulse toward the use of COTS products is primarily motivated by the DoD's desire to concentrate on its central, key business area—warfighting—and to rely on outside sources for other functions. The DoD has neither the time nor the resources to be a software company. Any attempt to duplicate the capabilities of commercial companies would focus too much effort on building software components rather than on defense, the proper business of the DoD.

*The poor general lives to fight battles.
The wise general lives to win the war.*

The use of commercial products can have profound and lasting impact on the spiraling cost and effort of building Defense systems, particularly information systems. It is important, however, to remember that simply “using COTS” is not the end in itself, but only the means.

*You must share your whole house with the cat
if you wish to be rid of mice.*

The entire process of system development—requirements, design, implementation, testing, maintenance—will undergo many radical changes when a system incorporates multiple COTS components. By committing to just one part of the “new paradigm” (i.e., simply buying some commercial products), it is vital to realize that this will usually demand a commitment to all of it.

*When the captain orders the horsemen to advance,
only the horsemen should advance.*

There are numerous policies that address the use of commercial products in DoD systems, but not all systems are subject to all policies. Before making any binding decisions about a system, it is prudent to verify which policies are in force.

*If the merchant sells you a painting, leave it alone. You cannot
make it more beautiful unless you are willing to become an artist.*

Modification of a product's code should almost never be considered as a viable approach when using COTS products. Note that this caution is not aimed at those products that are expected to require either parameterization or customization; this is often necessary for certain classes of COTS software. But a user who chooses to refashion a product in some manner not intended by its vendor will generally be following a dangerous course.

*The bullet travels farther than the arrow.
But the rifle costs more than the bow.*

The essential reason for a COTS-based approach is only partially to save money. The more fundamental reasons to use COTS components relate to such issues as rapidity of system deployment, timely maintenance, and ease of modernization, all of which are vital for a modern computer-based system. Thus, immediate cost savings from COTS may in many cases be minimal or illusory. It may even be true, especially in the short term, that a shift to use of COTS products can lead to higher costs.

加 时

The Shaping of Destiny in the Forest, the Forging of Victory by the Soldier

(Requirements and COTS products)

*We can only shape the path as we are cutting
its course through the forest.*

Many requirements of a system become known only as the system develops. This is especially true for a system that makes use of multiple commercial products, since their interactions will have a substantial influence on the system's eventual design.

*Our ancestors armed their soldiers with swords, we arm our
soldiers with rifles. But now we must buy them bullets as well.*

Some COTS products may themselves impose additional requirements on the system. Such "derived" requirements are usually unforeseen, but are no less important than the original requirements.

*Who commanded that twenty trees be felled?
We are only building a small raft.*

Requirements must flow from genuine needs, which must originate with the true stakeholders of the system. The simple fact that a requirement exists does not mean that it is truly necessary. Thus, even though answers may be difficult (or impossible) to determine, the following questions might be useful to ask: Who wrote that requirement? What is its source? Can its author explain the need that motivated that requirement?

*Have you ever lived in the forest?
Then how can you teach me how to survive in it?*

It is a certainty that any new DoD systems will be expected to incorporate COTS products where feasible. This means that when specifying the requirements of those systems, some awareness of the available products, their functional constraints, and even their design assumptions must be available to the authors of the requirements. Otherwise, there will be a basic and irreconcilable disconnect between the system requirements on one hand and the intention to use COTS on the other.

*In our ancestors' day they used old strategies and shot arrows.
Today we use new strategies and fire rifles. Do not use old strategies when firing a rifle.*

In a traditional “waterfall” process, requirements might properly come first. But different processes impose different constraints on development: when building a system from commercial products, requirements must be defined iteratively. In fact, many requirements will only become known after several prototypes have been produced.

*Soldiers need both rifles and blankets.
But they cannot fire blankets at the enemy.*

Not all requirements are equally important; there is more likely to be a spectrum from absolute requirements to desirable features to some things that are only marginally important. This implies a different approach to COTS product selection than simply verifying that all of the “shall” requirements have been met.

The forest fire needs a torrent to extinguish it, but the candle only a spit of water. Do not waste a torrent on the candle.

Some DoD systems have requirements all of which are precise and hard. In other cases, however, not all of the requirements need be as stringent as they have been expressed. It is often useful to validate a set of requirements: does the system truly need this feature, or capacity, or constraint? Is it absolutely necessary, or simply desirable? What would happen if the requirement is not met? Especially when incorporating several commercial components in a system, it is very possible that some requirements can be waived or loosened without substantially affecting the inherent functionality or performance of the system.

Last night we marched north through the forest. But now we have risen from the valley and can see the enemy camp in the west, so we have changed our course.

A complex system, even one with a well-defined purpose and function, has many aspects that cannot be known until users try it in a variety of circumstances: this is the very basis for using a prototyping approach. This means not only that many requirements can be left undefined, but that they must be left undefined until fairly late in the development process.

际 生

The Rifle, the Cook, and the Mountain

(COTS Product Evaluation and System Design)

The gunsmith must first measure the bullets if he hopes to build a rifle to fire them.

Designing a COTS-based system is inseparably wedded to evaluating the candidate products that will comprise it. Therefore, the evaluation process must start early in the lifecycle—in fact, at the very same point when the system is designed.

No one goes into the empty restaurant.

While the notion of what “commercial” means can be extremely difficult to pin down, an unmistakable symptom of commerciality (by any definition) is market share: the product’s vendor has sold it to several other users. Evaluating the extent of a product’s user base can be as significant as evaluating the nature of its functional behavior.

Who should you trust to guide you through the high mountain passes? A man who knows the mountains.

When evaluating a product for use in a long-lived system, one is implicitly deciding whether to enter into a long-term relationship with the product's vendor. That vendor's stability, business acumen, and commercial viability may be as critical to evaluate as the inherent quality of his product.

There are no perfect weapons, so do not look for them. Buy what weapons you can find, and use them as perfectly as you can.

A commercial software product will almost never fit perfectly in a complex system (e.g. in the sense that it can easily “plug-n-play”). Such a perfect fit can only occur if the system is explicitly designed around that product. Therefore, some degree of misfit is inevitable, and should be expected. The key for success is to evaluate the misfit in advance, since then it can be seen whether it can be accommodated.

The cook in the restaurant offers either rice or soup. Which is better?

Evaluation of COTS products often must contend with comparing products that are not entirely comparable. The evaluation criteria must therefore be rooted in the characteristics and needs of the system that will use the product, not solely in the functional characteristics of the products.

How do you know how high the mountain is? Have you climbed it?

One thing that is almost mandatory when evaluating a COTS product is to get it in hand, try it out, and ascertain on a first-hand basis whether it is fit for the intended purpose. The extent of this activity is related to the importance and cost of the system, but at least some hands-on evaluation is necessary.

If you cannot load your rifle, it is no better than a walking-stick.

Evaluating the quality of a COTS product is not found solely by examining its technical workings. Evaluating the quality of the product's user manuals and documentation, and the quality of the product support are as important as, and sometimes more important than, evaluating the product's functionality.

The cook will always praise his own food.

A COTS vendor is in business to sell products, and his literature will naturally tend to stress his product's qualities. But it may also include ambiguous, or even misleading statements about the product's aptness for one or another purpose. Assuming that choosing a particular product for use in some system is an important decision, then it is unreasonable to make such a decision based principally on a vendor's literature.

One man sees the mountain through the branches of trees in the forest, another sees it reflected in the water of the stream. Neither sees the mountain as it really is.

No matter what evaluation methods are used, it will almost always be impossible to determine the absolute quality of a COTS product in an objective manner. This is because the value of a COTS product—its fitness for use—is entirely wedded to the system context in which it will be used.

*At first, the soldier picks up the rifle to shoot it.
But in the heat of battle, he may use it as a club.*

It is sometimes difficult to predict the way that a COTS product can bring its own assumptions on usage to a system. Introducing one product can influence the way other components are integrated, or the system's overall design, or the processes that the system supports; this is a natural by-product of a COTS approach. It can be a useful strategy, especially for system designers, to regard this serendipity as a source of potential value rather than as a drawback or limitation.

盟 谊

The Horse, the River, and the Sea

(Maintenance of COTS-based Systems)

No river keeps to the same course forever.

The evolutionary path of a COTS product is only partially driven by technical reasons; equally important are market pressures, competition, and the vendor's strategy. If a vendor decides to make significant changes to his product, then any system that uses that product will be affected accordingly.

The rider must stop to change horses on a long journey, even if he is in haste.

New product releases occur periodically, and products are sometimes discontinued. If a system makes use of many COTS components, maintaining it means upgrading its components regularly. The resultant overhead, whether measured in downtime, reinstallation costs, or anything else, is a necessary by-product of a COTS-based approach.

When the river changes its course, we must rebuild the bridge.

One common model for a COTS-based system is a collection of commercial products held together by “glue” code. However, commercial products change and evolve over a system’s lifetime, and the “glue” that holds a system together will also need updating, evolution, and maintenance.

If I own a horse I must feed him.

If I use my neighbor’s horse, the horse still must eat.

Choosing a COTS-based approach does not remove the cost of system maintenance, but rather shifts the cost to a different locus. Now, instead of paying a team of programmers to repair bugs, one pays vendors for product updates, and integrators to write new encapsulations and glue code.

*The passengers do not choose the time to sail;
the ship departs when the tide is high.*

Maintenance of a COTS-based system involves replacing its components with updates and new releases. The schedule of those new releases is determined by the vendors of those components; the system’s maintenance schedule will therefore be, to at least some extent, dependent on the vendors’ release schedules.

乐 新

The Boat on the Great River

(Business Processes)

It is foolish to board the boat if you do not intend to cross the river.

Certain classes of COTS products will assume that their users subscribe to some particular business processes. Making a commitment to use such a product therefore requires a parallel commitment to adopt the processes the product supports, regardless of whether they conform to your existing processes or not.

When the boat is at the pier, the water is shallow and no one is afraid. But in the middle of the river where the water is deep, everyone is fearful of drowning.

Many organizations express their willingness to reengineer their business practices when beginning a project to introduce COTS products into their systems. But when the full implications of the reengineering become known, that willingness can sometimes disappear, often in mid-project and when a considerable amount of money has been committed. This is the wrong time to realize what “business process reengineering” really means.

The helmsman may move the rudder, but the whole crew must help reset the sails.

An organization's processes are embodied by its personnel. A decision that business processes will be reengineered to accommodate a commercial system is actually a decision that all of the organization's people will adjust their daily activities. Countless bitter experiences have shown that people do not change simply because an edict is made, but through education, training, persuasion, motivation, and leadership.

*Some boats have one sail, others have two.
Which one should you take when the storm is rising?*

Different COTS products exhibit different degrees of process flexibility. The choice of a product for use in a specific context must be based not only on the essential business processes that the product supports, but also on a realistic assessment of whether—and at what cost—the product can be tailored and customized to a particular set of needs and circumstances.

节 天

The Farmer is Heroic, the Blind Man is Wise

(Testing and Debugging)

Even if the blade breaks, the farmer must somehow plough his field.

When a COTS-based system fails, the failure affects only the owner of the system, not any of the vendors that made its components. And therefore it is the builder of the system who must deal with that failure, by identifying the point of failure and incorporating workarounds.

The beautiful orchard can bring forth sour fruit.

Even reputable COTS vendors can produce products that have defects. All components of a system, therefore, regardless of their origin, must be tested with the same rigor that is applied to custom code.

The cattle cannot speak: it is hard to know what ails them when they cry.

When systems that contain commercial products fail, it can be very difficult to detect the cause of the error. The source code is not available, nor are the design assumptions and constraints that guided the developers of the products.

*The blind man learns to rely on all of his other senses.
Only through them can he learn to see again.*

It is seldom possible to “debug” a COTS component, since it is effectively a black box to its user. A system that uses COTS products, therefore, can only be debugged by observing the system’s behavior, and thereby making inferences about how the components work. This is a very different task from debugging by means of source code, and is often considerably more difficult.

*The anxious farmer brings his crops to the market early.
How can he do this? By picking them too soon, when they
are not yet ripe.*

COTS vendors are under enormous pressure to release products as quickly as possible. This implies that products are often released with only minimal testing and debugging. This circumstance stems from the realities of the commercial marketplace, and will not change in the foreseeable future.

*Do not blame the corn I sold you; it was some other
farmer’s rice that has soured the soup.*

When several COTS components interact in a complex system, and the system fails, it is frequently very difficult to determine the actual cause of failure. Even if the vendors offer extensive product support, it is often true that each vendor will place the blame on someone else’s product, and no help will be forthcoming in determining the cause of failure.

节 天

Glorious Victories Will Come From Harmonious Workers

(Integrated Product Teams)

Farmers swing the scythe in great arcs when they are harvesting wheat. But they are wise not to do this in the apple orchard.

One set of management practices cannot be regarded as equally appropriate for both traditional and IPT approaches. Deciding to execute a project with integrated product teams implies that the organization's management processes will change, perhaps radically, to accommodate some very different needs and constraints.

Do not fire at your own soldiers.

Contractors and Government members of an integrated team are on the same team, not on opposing teams. While the contractor and Government persons represent different organizations, it is still necessary to assume that all members of the team share common goals.

*The boat takes many oarsmen to row it.
But someone must sit at the helm.*

IPTs provide a useful modularity in system development. But it is still necessary for some central authority to make overall decisions about design, integration, and overall strategy. Someone, regardless of job title, must perform the role of the chief architect.

The wise captain already knows what the messenger will tell him.

A team that is truly integrated has little need for reviews of the traditional kind. One major indicator of whether this kind of integration is present is that the team's Government members should be as familiar as its contractor members with the ongoing status of the project. If they are not, then it is doubtful that the team is an IPT except in name.

*You travel on the river by rowing, but on the land by marching.
You cannot do both at the same time.*

If some teams on a project use a rapid prototyping approach and others fall back on defining all of the requirements upfront, this is probably a symptom of a mismanaged project. IPTs cannot be totally independent, and one key responsibility for the project's technical leadership is to look for such symptoms and negotiate an approach that is common to all teams.

Your cousin lives far away; how can he help to harvest the wheat?

The notions of “team” and “integrated” imply a certain amount of physical closeness, and a successful IPT will very likely be one where physical proximity is a given. Even granting the technologies that now make “virtual teams” possible, there is abundant evidence (borne out by common sense) that a distributed collection of people that constitute a team will have a more difficult time than one that is physically co-located.

加 时
联 行
明 由
民 迎

How will the Great Revolution succeed? By each worker reading the Little Red Book and taking its message to heart.

The success of the new paradigm—extensive use of COTS products, flexible and efficient development methods—is directly related to how every individual in the community adjusts and changes his or her day-to-day work to accommodate it. Without this personal commitment, the “new paradigm” will be nothing more than a policy without a constituency, and the only true aphorism will be “Business as usual.”

Acknowledgments

Several friendly Comrades made valuable comments on these quotations, and I would like to thank Kurt Wallnau, Patricia Oberndorf, Scott Hissam, John Foreman, and Terry McGillen for their insight and good will. Mark Paat, Jamie Vorhes, and Barbara White were extraordinarily helpful with formatting and publication. And special thanks to Ed Morris for helping make the pithy sayings pithier.



CarnegieMellon
Software Engineering Institute

For More Information, contact—

David J. Carney

Phone

412 / 268-6525

Email

djc@sei.cmu.edu