ph: 802.355.0430 email: ARTS@LOUDCANVAS.COM
website: WWW.LOUDCANVAS.COM

LOUD CANVAS MEDIA
13 BACK RIVER ROAD
DOVER, NH 03820

On The Integration of Lean Principals with a Pragmatic I/T Security Model

By Sean Dempsey
October 10, 2008

In today's rapidly changing I/T environment, security seems to be transitioning from 'an important after-thought' to a real market differentiator. The choice whether a business invest in a proper security model or not can now mean the difference between gaining new clients and increasing market share or becoming obsolete. However, is the current buzzword "security" being under- or overplayed in organizations? Can there be "too much of a good thing" when it comes to security or is the more the better?

Most importantly, is the amount of capital invested in creating and maintaining a proper security model directly proportional to increased sales and new lines of business? In a nutshell – does spending more (than the competition) on security truly increase your appeal to consumers? If so, is there a point where there is "too much" money spent on security and not enough on other aspects of the business? How do we gauge these issues?

This paper attempts to candidly grapple with some of these issues. Many of the points to be covered are based on objective reasoning; others on subjective thought and experience. Feel free to read along and question the thought process as you see fit.

Lean Thought Process

Lean methodology (of the ever popular *Lean Six Sigma*) informs us that the cornerstone of successful business is to provide only what the customer wants and is willing (and able) to pay for. Thus, a successful business only provides and performs processes that add value for the customer (*Womack and Jones 2003)*. If the customer doesn't want to pay for it, it shouldn't be done. It's a very simple concept, really. Why "do" something that the customer doesn't think is important…?

As a result of this thought process, great strides are taken (for companies that embrace this mantra, such as Toyota and Motorola) to eliminate what customers are not willing to pay for. If the customer wants to get from point A to point E, no need to take him to points B, C, and D along the way. Such process "steps" or "activities" are often called *non-value added* (or *waste/muda,* depending on the book you read).

Providing Value – A Simple Example

Let's—for example—consider we're running a business which offers emotional and psychological counseling. We've determined our customers/clients are willing to pay for one and only one thing: the provision of counseling. Now we could offer additional services – hot towels to put over their faces while they lie down, free candy, cheery balloons at the end of each session – but remember the customer is only *willing to pay for counseling*. We can offer these "accoutrements" but if they cost us money (and we charge the same) we are only eating into our profits. If we charge a premium for adding

ph: 802.355.0430 email: ARTS@LOUDCANVAS.COM
website: WWW.LOUDCANVAS.COM

LOUD CANVAS MEDIA
13 BACK RIVER ROAD
DOVER, NH 03820

these services, we risk losing the customer's business (since he/she might go to a cheaper counselor who offers only what they're willing to pay for).

These same concepts apply for underlying processes that take place behind the scenes. If we spend thousands of dollars installing a fancy phone system or increase the size of our parking lot, have we added value to our customer? No. Why? Because the customer has not stated they have a problem with our phone system or our parking lot. They are ONLY willing to pay for counseling. Why add (and over complicate) a very simple need?

To quickly restate -- all the steps in the process that directly relate to providing the customer with counseling are those that **add value**.

Extending the Example
Now this article is about security. Don't worry, we'll get to that. However, let's backup a bit and next consider a few to our customers' unspoken needs:

Our customers expect to be treated with respect; they expect us to be professional; and they expect us to maintain their privacy and keep their emotions confidential. In addition, they may have a slew of other needs that are not explicitly stated but assumed.

Now – here's the question: Do these customers expect to pay a premium for these "assumptions"? They expect respect; should they pay extra for us to **not** laugh at their issues? They expect professionalism; should they pay extra for us **not** to conduct counseling sessions in our underwear? And they expect confidentiality – should they pay extra to have us **NOT** leak their deepest secrets to their friends?

It is my belief that the answer to these questions is no. Why? Because all these examples (respect, professionalism, and confidentiality, etc) fall within the lines of what we define as "to provide counseling." We are in business to provide a service, not to NOT do something. The customer should not pay for us to NOT do something – but the opposite.

These assumptions—unspoken or spoken—thus contribute to us providing our service. We need them to conduct our business. They can be referred to as non-value added but necessary elements in providing our service (Baudin 1996). There are many examples of these types of activities—and this is the focus of this paper. Insomuch, I'll be getting into more specific "I/T" examples of such activities later.

More *non-value added but necessary* activities in providing counseling would be the following:

- Obtaining fire insurance to insure the building
- Taking time to properly categorize and maintain client contact information
- Doing background checks on employees before hiring

In effect, any activity, step, or sub-step in the process of providing our service that the customer is not willing to pay for BUT we must do (or feel we must do) to stay

ph: 802.355.0430 email: ARTS@LOUDCANVAS.COM
website: WWW.LOUDCANVAS.COM

LOUD CANVAS MEDIA
13 BACK RIVER ROAD
DOVER, NH 03820

competitive, mitigate risk, or prevent losses is **non value added but necessary**. They are nothing really but "necessary evils"

But how do we approach and handle these types of activities?

Extending the Example
So we now know some tasks do not intrinsically add value—but we must (or should) do them nonetheless. Thus the question becomes HOW MUCH of these implicit aspects of our business do we have to focus on and invest financially? Is there TOO MUCH of a good thing—like security…?

Let's consider one last time our counseling business. We know that the world is not perfect (if it was non-value add steps would not exist). We also know corruption is prevalent and persons may try to sneak in and steal our confidential customer information. We can mitigate this risk by installing a security system at our office and a safe for our customer data.

Now we're talking an investment… an investment in "peace of mind" (that we will not face a breach of confidential data and potentially lose customers as a result). But how do we know how much to spend on this security? There are so many types of safes…so many brands of alarms and locks. And do we get "the best and brightest" to install these systems. If we do … then we can loudly proclaim that our customer's information is safeguarded by the best! Perhaps we could increase business by doing this…

But why not go even farther? We can hire security guards to sit and only admit those with valid appointments. We can encrypt our customer files digitally so the only way anyone can see pertinent information is to use a special decoder ring that only the CEO has and has to signoff on its use. Then we can provide training on the new processes we've defined – such as how to decode all necessary information enabling us to simply call a customer and setup an appointment.

As you can see this example has gotten a little ridiculous; but the point should be obvious. How far is too far with providing security for our customers? We've added "security" but at what cost? We could potentially add so many non-value added steps to the process we lose track of our real focus and business objectives.

But, conversely, what is "not enough?"

Why Invest in "Waste"?
"Business 101" tells us that in order for any business to stay competitive, it has to effectively balance its finances, time, effort, and services. In a nutshell - tradeoffs exist. Logically if we're investing $1000 in equipment, we don't have that $1000 to spend elsewhere. More important—for the sake of this discussion—if we invest in **security** we're NOT investing in marketing, infrastructure, new buildings, financial investments, or paying (as many) dividends to stockholders.

ph: 802.355.0430 email: ARTS@LOUDCANVAS.COM
website: WWW.LOUDCANVAS.COM

LOUD CANVAS MEDIA
13 BACK RIVER ROAD
DOVER, NH 03820

Why do businesses invest money in any of these? Short of making this a paper on finance, the business' main objective is to maximize profits. The business invests capital to maximize near and/or long term profits; it does this by increasing gains OR decreasing the amount of losses. Investment in non-value added processes and steps like **security** are for the latter – to mitigate risk and DECREASE amount of losses (ideally make them zero).

How Much to Invest?
So then -- the BEST number we can hope to strive for (in losses as a result of our investment) is zero ($0). We cannot directly "gain" money by investing in security—unless, as mentioned in the first paragraph, doing so differentiates us enough in the industry to glean further business. However, by and large, we are often **not** able to directly correlate new business lines to investment in security.

As such, at our very best we can never hope to "lose" the company **less** than $0 as a return on our investment. Less confusingly said, we can rarely "make" the company money by investing in security. Therefore, ANY money invested in security can be treated as pure expense and the ROI will (almost) always be -1.

The formula describing this relationship is described as following:
ROI = (Gain from investment – Cost of investment) / Cost of investment

(since gain from investment can at most be $0):
ROI = -Cost of investment in security / Cost of investment in security

(so assuming a 200mil investment in security):
ROI = -$200 / $200 = -1

From a purely ROI perspective, it should be obvious that the least amount of money we can *effectively* spend on security the better. The "investment" paradigm that currently is held when considering security is debunked. **Security is not an investment**, it is purely an expense. Thus – as with any other expense the company should spend the LEAST AMOUNT OF MONEY POSSIBLE to provide adequate levels of security.

Furthermore, the analysis shows security fundamentally does not differ from any other expense. What is notable about it is that is falls into the same category as "insurance." It is an expense for purposes of risk mitigation, but does not contain the formal accounting category of *intangible asset* as insurance does. Perhaps it should in theory – but in business reality it does not.

Like insurance, it would be illogical to "invest" any more than absolutely necessary in security to ensure full coverage. The "bare minimum" should be strived for rather than the "absolute best." It is this change of thought process that must take place in organizations—as it aligns with Lean concepts and a customer-oriented focus. The company must pay only the dollar value that provides it with full security. Going "over and above" does the opposite effect on profits than desired (both long and short term).

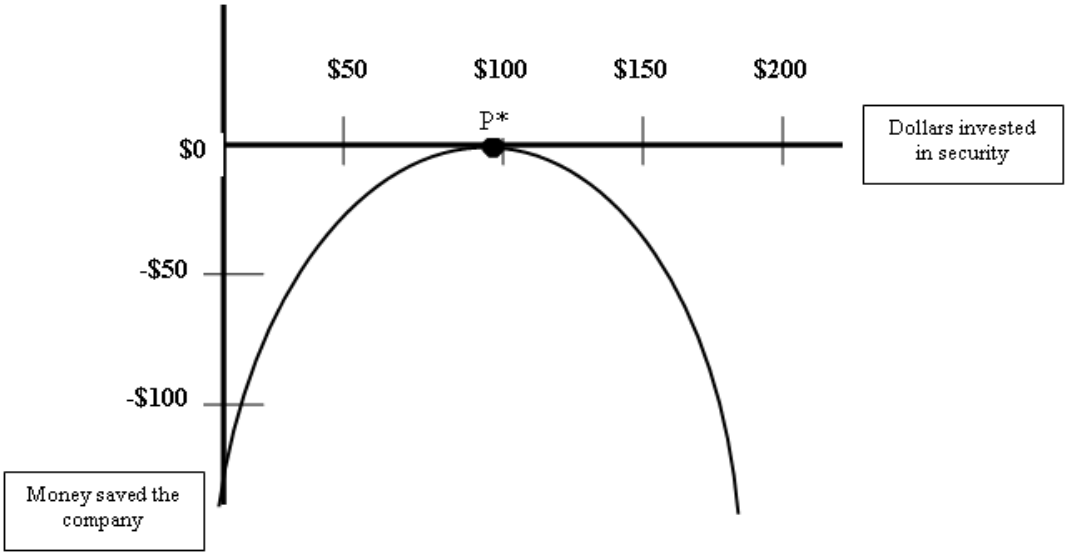
ph: 802.355.0430 email: ARTS@LOUDCANVAS.COM
website: WWW.LOUDCANVAS.COM
LOUD CANVAS MEDIA
13 BACK RIVER ROAD
DOVER, NH 03820

Investing more than you need to be fully secure is the same as paying more for insurance than you have to.

To demonstrate this point, the below diagram describes the inverse quadratic relationship between dollars "invested" in security (x-axis) and money saved for the company (y-axis). As you follow along the curve (from $0), you decrease the theoretical amount of "loss" you might experience with a data breach. However, at the apex (pinnacle), if you spend one marginal dollar more you are spending more than is necessary for the optimal level of security.

Thus, the ideal level to seek is P* which is the minimum number of dollars invested to give the maximum [neutral] "return" of $0.

Figure 1- Quadratic Relationship



Of course the numbers are purely demonstrative. $100 could just as easily mean $100,000 or a 100 million depending on the size and maturity of the company.

The other thing to consider that is that these "investment dollars" (or more accurately "expense" dollars) can be on a cyclical basis. Many large companies are spending millions of dollars *a month* on security—due to ongoing and costly procedures in place and/or personnel cost associated with maintaining those procedures/policies.

ph: 802.355.0430 email: ARTS@LOUDCANVAS.COM
website: WWW.LOUDCANVAS.COM

LOUD CANVAS MEDIA
13 BACK RIVER ROAD
DOVER, NH 03820

Concept Applied / Comparison to Agile Development
How do we achieve this "maximum" return for the "minimum" price? As you are no doubt aware, the ramifications for not 'meeting the mark' can be disastrous. A security breech that could have taken $100 to fix might cause tens of thousands of dollars in lost revenue from lawsuits or lost consumer faith.

Truly, there is no "magic formula" to calculate the optimal levels of compliance and security policies to put in place for any company. The appropriate and "adequate" levels are dictated by context. (University of Miami 2005). For one thing it is largely dictated by the I/T maturity model of the organization. On a whole, companies with large numbers of employees and an I/T architecture that is not completely transparent will have to spend significantly more on security than smaller companies. In addition, companies whose business models are driven by the handling of sensitive customer data (credit cards, social security numbers, etc) will be regulated by state and federal mandates; their internal (and external) controls must be tighter.

However, we can step back and analyze the bigger picture and get down to core concepts. We've addressed that security should be applied as a risk mitigation expense and, as such, at the least cost possible to fit the need.

If you've worked in software development before, this concept might sound somewhat familiar. In keeping with Lean principals, the recent **Agile** methodology has espoused a new archetype for handling aspects of software development—cutting down the "non valued" steps to the basest elements. Namely, *documentation* (when unable to be cut out completely) is written per the "least amount" for the "maximum value." The focus is on communication, succinctness, and being "lightweight." The goal is to write documentation that is "just barely good enough" (Ambler 2008).

It is with this same mantra that security should be addressed! Oh, when stated explicitly it sounds like a recipe for failure. Security that is "just barely good enough…" However, it is only by addressing the "expense" of security in this way that the business does not over compensate and swing on the right side of the pendulum (the quadratic equation).

As a quick caveat—given the fact that a security breech is often far more serious than not correctly interpreting software documentation—I believe that the mantra should be defined as "security that is ***good enough*** to do the job."

More specifics / Security Defined
Let's get into some more specifics regarding "how" to provide maximum security for the minimum price-tag.

For that matter, I believe I have not even broken down what security means from Information Technology standpoint. Often referred to as *CIA*, security can be broken down into 3 base elements: ***Confidentiality***, ***Integrity***, and ***Availability*** of Information Assets. (Adrian 2005).

These three types are defined thus:

ph: 802.355.0430 email: ARTS@LOUDCANVAS.COM
website: WWW.LOUDCANVAS.COM

LOUD CANVAS MEDIA
13 BACK RIVER ROAD
DOVER, NH 03820

> ***Confidentiality*** refers to limiting information access and disclosure to authorized users -- "the right people" -- and preventing access by or disclosure to unauthorized ones -- "the wrong people."
>
> ***Integrity*** refers to the trustworthiness of information resources. It includes the concept of "data integrity" -- namely, that data have not been changed inappropriately, whether by accident or deliberately malign activity. It also includes "origin" or "source integrity" -- that is, that the data actually came from the person or entity you think it did, rather than an imposter.
>
> ***Availability*** refers to the availability of information resources. An information system that is not available when you need it is at least as bad as none at all. (University of Miami 2006)

Let's look at each, one at a time.

Confidentiality
Confidentiality deals with who has access. If we can drill down now and talk specifics for I/T, this means access to Production data (and code). Only those who should (the users) have access to Production should have access.

The drawbacks of implementing "confidentiality" security become apparent immediately. Do we revoke access to production from those developing and supporting the application (production support/developers)? How will they fix errors that must be handled immediately? How will they troubleshoot issues?

Yet if we don't revoke their access they could do any level of harm in the system. They could download everyone's social security and credit card numbers to a disk-drive; they could introduce malicious code that steals customer files or sends mass spam emails. Can we balance these two extremes?

One popular solution (often executed in many large, I/T mature organizations) is to implement ***Role Based Access Controls*** (RBAC) in order to ensure that those developing code for production do not have ability to deploy code to production. (see Figure 2 visual below)

ph: 802.355.0430 email: ARTS@LOUDCANVAS.COM
website: WWW.LOUDCANVAS.COM
LOUD CANVAS MEDIA
13 BACK RIVER ROAD
DOVER, NH 03820

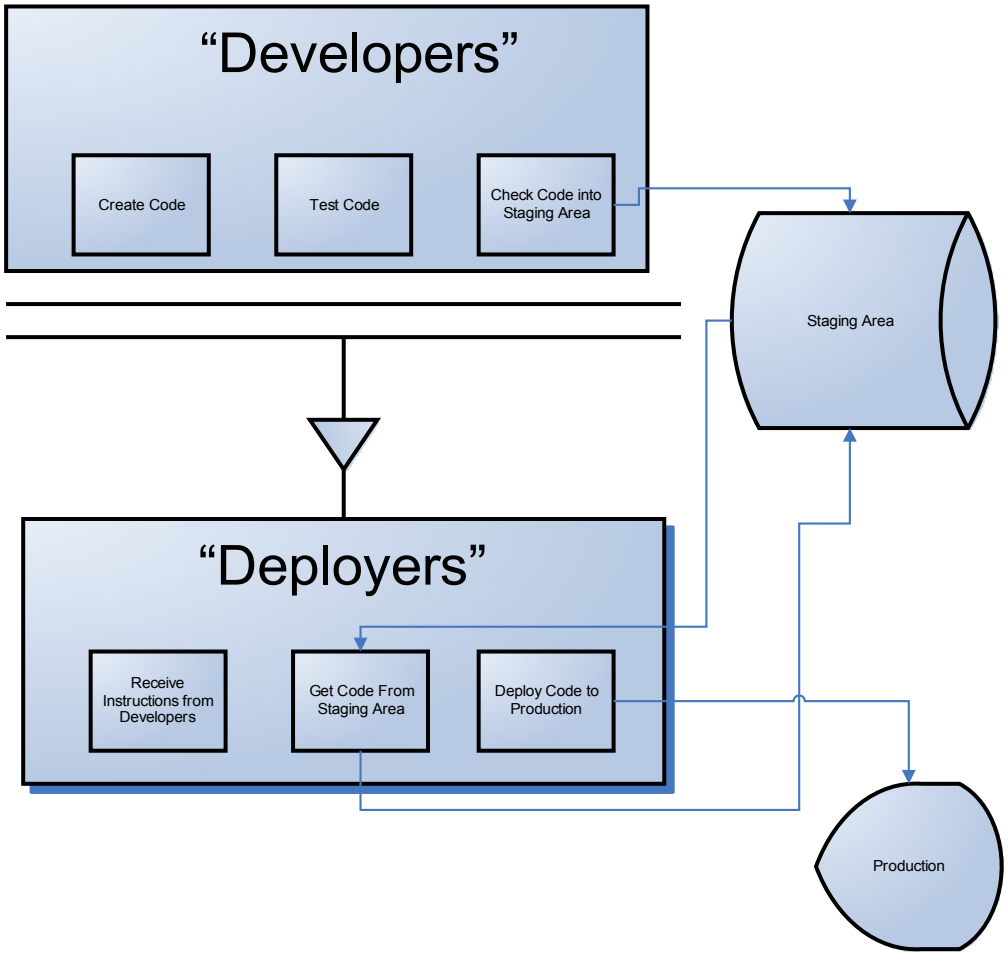


Figure 2 - Implementation of RBAC

It is quite evident that this process is inherently not "lean" by its very definition. It is requiring 6 steps and at least two people to do the same job as one person and three steps. It has effectively doubled the amount of work necessary to do the same job; this creates *muda* (waste) and overcomplicates a simple process. Moreso, if there is any "lag time" in the communication between these two roles, the cycle time will be increased even more.

As such we now have gone from a fairly straight forward process of "Create Code" → "Test Code" → Deploy to Production to a jumbled mess that requires extra coordination, additional controls and procedures, more people, more interfaces, more time, etc. etc.

Yet via this process we now have "security". The risk of confidentiality being exploited has been mitigated. But even now some questions still remain…What's to stop the developer from simply writing "bad code" and staging it for the Deployers to promote? Nothing (via this model). So now we need additional provisions to keep that from happening (code reviews, more security checks, id management and controls, etc). I will

ph: 802.355.0430 email: ARTS@LOUDCANVAS.COM
website: WWW.LOUDCANVAS.COM

LOUD CANVAS MEDIA
13 BACK RIVER ROAD
DOVER, NH 03820

not go into the lengthy and complex process that must be created to enforce this pattern of separation of duties. Needless to say, however, the non-value added steps should be apparent.

As a caveat -- there is indeed something to be said for the separation of duties: when exercised correctly, it is very effective. The ability for one person to act unscrupulously is only limited to the persons integrity and availability to the information. Having two (or more) persons work in conjunction with each other to be illicit is far more rare.

Yet, at what cost is this new process effective? Hypothetically if it took 25 million dollars to implement the security model defined with a mere ***potential*** to "save" the company a loss of 10 million, we have effectively spent 25 "REAL" dollars to save an "IMAGINED" 10. Obviously some cost/benefit analysis needs to take place when considering and balancing these alternatives.

I propose a revised process that embraces (not discards) Lean methodology for enacting Confidentiality security. The former process (above) describes the belief that the organization must "control" the deployment process and demand compliance. It is an expensive and dogmatic approach. Instead, why do these organizations not condone a philosophy of ***Accountability*** rather than ***Control*.**

Specifically - those that promote code into production should have production access (to do their job) *but* only do so using their own identity (id). If we're talking a Unix/Linux environment, no ids would have access to the box other than root (for the application – and no one would know the password) and the developers own ids. There must be no "generic ids." This would ensure that those promoting code, fixing issues, and "logging in" were accounted for. Their actions and identities could be traced (if issues arose). The reasoning for this is simple: it is widely believed that accountability should prevent ***the SAME*** number of incidents as direct control and oppressive, preventative measures.

This method keeps the number of steps reduced, maintains flexibility, promotes responsibility, and keeps the model of security "just good enough" rather than throwing effectiveness and cost-sensitivity out the window. In addition, it promotes a spirit of "faith in the employees" within the organization, rather than fear and distrust. This ideology, in and of itself, lends itself to a more effective and dedicated I/T workforce.

Integrity and Availability
Without going into as much detail as above, the same Lean concepts carry over for integrity and availability of data. Truly, "less is more" when dealing with the processes to put in place in order to ensure integrity, reliability, and availability of information.

As availability can often deal with such security issues as "denial of service" attacks and other malicious external user assaults, the internal controls are not as relevant. However, the importance of "writing secure code", for example, and coding the application to deny attacking IP can be crucial.

ph: 802.355.0430 email: ARTS@LOUDCANVAS.COM
website: WWW.LOUDCANVAS.COM

LOUD CANVAS MEDIA
13 BACK RIVER ROAD
DOVER, NH 03820

What's important not to do is to spend *unnecessary* amounts of money and time on attempting deal with incredibly remote security attacks. For example, when working on applications that exist within an intranet or behind a firewall or proxy, spending weeks restructuring all existing code to counter possible SQL injection or Javascript injection may be a waste of time. If the application is externally facing, however, and users may attempt to hack the system, these preventative measures are incredibly important.

Likelihood of attack and from what *direction* must be taken into account when coding security. To not do this is like spending all day boarding up the windows of your house when you've left the front door wide open. Focusing on the important and most probably sources of breach take precedent.

Finally, the sheer number of unnecessary hours spent on security is often the result of poor planning. Security in I/T applications cannot be an "afterthought" it must be integrated into the solution and present from day one. If security is attempted to be "patched" over the application after it's been developed, it will 1) be far less secure, and 2) will take 10x the number of hours had the effort begun at the start.

Security should be fundamentally apart of the initial requirements for the software being developed.

Conclusion

This article was meant to address what the author perceives as the misaligned and false paradigm of "Investing in Security." It is his belief that security is not an investment or an asset – and does not ***inherently*** contribute to customer satisfaction. It is a non-value added but necessary step in a process of providing the customer a good or service. Thus it must be treated as an expense and only the least number of effort hours, time, and finances be devoted to it as is needed. The mantra "a security model good enough to do the job" (dictated by the organization) should be assumed. Lean principals apply to it just as any other step or activity in the process for providing the customer.

About the Author

Sean Dempsey works part time as an I/T consultant and developer. He has worked with more than 30 small businesses and non-profits, consulting them on issues such as security and web standards. He works actively to create and improve client web sites and develop effective web solutions. He has a Bachelors in Business and a Minor in Computer Science from the University of Vermont. He is currently perusing his Green Belt in Lean Six Sigma as well as his Masters in Business at the University of New Hampshire.

ph: 802.355.0430 email: ARTS@LOUDCANVAS.COM
website: WWW.LOUDCANVAS.COM

LOUD CANVAS MEDIA
13 BACK RIVER ROAD
DOVER, NH 03820

Sources / Bibliography

ROI for Security in Information Technology
http://www.geocities.com/amz/ROISI-Paper.pdf
Mizzi, Adrian. 2005

Value-added versus non-value-added tasks: a useful distinction?
http://www.mmt-inst.com/vavsnonvatasks.htm
Michel Baudin, 12/16/96

http://www.csoonline.com/article/446017/Separation_of_Duties_and_IT_Security
http://csrc.nist.gov/rbac/Role_Based_Access_Control-1992.html
Separation of duties

http://www.arcert.gov.ar/webs/textos/secure_webdev-3.0.pdf
Best online security practices

RBAC (Roll Based Access control)
http://en.wikipedia.org/wiki/RBAC

http://www.agilemodeling.com/essays/agileDocumentation.htm (Agile Development)
Scott W. Ambler

Security Defined / CIA
http://privacy.med.miami.edu/glossary/xd_confidentiality_integrity_availability.htm
Unverisity of Miami 2006