Recommendations and Suggestions for CRE Loaded

Written by: Laura Wheeler

- Small Business Web Development Professional and Expert
- Principal Owner Firelight Business Enterprises, Inc, Medicine Bow, WY.
- Instructor and Curriculum Writer for Small Business Web Professional Certification Program, University of Wyoming

The information presented on the following pages is given with consideration for both a sustainable business model, and for recovery and enhancement of the CRE Loaded Open Source community. This is written as a series of personal recommendations. As such, I hope you will excuse my sometimes vehement enthusiasm. I feel very strongly about much of this. Please excuse my bluntness on some issues also – without it, I cannot hope to explain why change is needed, or how the change can help.

Until the release of version 6.3, CRELoaded maintained a strong, though never stellar, position in the Open Source world. It lacked the fanatical and explosive following of projects such as Joomla and Drupal, and was not commonly known among end user market bases, and was only slightly known within the web design community. Most who knew of it were still unaware of the elements which set it apart from the other available options for shopping carts.

The mentality of the average user is not one of logical assessment. Very few web designers actually load and test cart after cart to do side by side comparisons on setup and maintenance efficiency, or features (I am one of the few who has done so – about 30 or more of them). They trust the feature lists that are published by the developers – often a far cry from the reality of using the software.

Judgments and recommendations are often made based more on familiarity than on actual performance – if they got the thing to work, in a near approximation of the need, they pat themselves on the back and smile at their accomplishment. Then, because the learning curve was so high, they fear learning another, based on the assumption that they all must be this difficult, this cumbersome, this awkward or time consuming. **They then build the next one in the same thing, no matter how hard it was!**

Very few think that there might be an easier option. **Individual store owners NEVER think about sustainability.** And web designers hide that little wrinkle from them, often building a cart and walking away, leaving it to age into unacceptable security risks, or coming to the “rescue” of the client when updates and patches need to be installed – often a nightmare because of the number of core files that must be edited to get the thing operational to start with. **It is only those who begin to build carts in high volume who understand the need for fast site build functions, and easy updates long term. This was our reason for selecting CRE to begin with – it is not what we want, but it is the best option available.**

It is also important to understand that many of the designers within the user base do not understand code – not the language, not the theory. They do not understand how code is structured, that there is good structure and bad, they do not understand what makes code forward sustainable, and what impedes it. They just want what they want, and expect it to be delivered. End store owners who do not even understand basic web functionality well, are even worse. They just want it to work, the way they want it to work.

This means that most of your end users, with the exception of coders, have no idea of what the real strengths are, or what the hindrances are to progressing better. There is no one to tell them what the strength is (since they don't really find it themselves), and they lack the comprehension or interest to learn why things don't move forward quickly.

So CRE, being only marginally better in certain areas, and not having a market focus on its REAL strengths, or a focused development path to strengthen them further, has sort of wallowed along, never quite competing on its true merits.

CRE Loaded possesses some advantages over the competition:

1. While templating is by no means convenient, it is far more consolidated than other contenders such as

X-Cart, and even the up and coming Magento (a MAJOR Achille's Heel for Magento, and a complete show stopper for us for that cart).

2. Updates can be a bit of a pain, but CRE has, to date, removed most of the elements which require code hacks and modification of core files to get the store running. Many other Open Source carts have not managed to get even that far.
3. **The functionality of the system, combined with those two factors, offers very small businesses and growing businesses a chance to get a cart that is less of a hassle to set up, and easier to maintain long term.** This presents tremendous cost savings to businesses in both the short and long term.
4. While much of the code base is aged and clunky, the cart itself is at least mature enough to have some stability, and a good range of desirable features.
5. **The strength has never been in the add-on modules. It has been in the flexibility of features, the beginnings of simplification of the build and maintenance processes, and in the ability to upgrade to more feature rich options for increased needs.**

Because the advantages have not been clearly defined, and because those advantages were not really tightly focused as the points of improvement, CRE has sort of coasted along, with designers seeing the potential, but feeling like it has never quite been realized as it should be.

From the outside, it feels like the project lacks strong direction, purpose, and targeting. It seems to bounce from trying to fix the big stuff, to throwing out little tiny improvements just because they were easy. It feels a bit unstable, and erratic, to the customer base.

Additionally, the dependence on the OSCommerce community for enhancements has crippled the cart, impeding progress forward in coding issues, and it has also meant that CRE could not develop a strong and enthusiastic following of its own. **The dependency provided short term advantages, but a long term handicap that has been strangling.**

Progress to date has been sort of hit and miss. Again attributed by me, to the difficulty of finding ways to seamlessly break from OSCommerce without pulling out the underpinnings that users have come to depend upon. Others attribute it to lack of dedication on the part of the team members, or lack of funding, or lack of understanding, or to many other much less charitable causes. **No accusation here, that is just how things are perceived by those who do not understand how software is developed.**

This seemingly harsh assessment is not a criticism of the developers – I acknowledge and appreciate the endless hours put in to get things as far as they did get – Open Source is VERY difficult to get to the point of usability and any degree of profit, so I know how hard you had to work to get it there. I also understand how difficult a project is to keep moving forward, and how demanding it is. It is just an assessment that perhaps the needed focus was not there, due to an unclear concept of what the end goal should be (where the real money was), who to listen to, and how to best get from point A, to point B.

The recent changes were certainly made in an effort to drastically change the patterns of inefficient progress. It is my opinion that slightly different strategies, if implemented would provide a stronger base, and a better forward momentum for the CRE Loaded project. It has careened from one extreme, to an equally unsustainable extreme in another direction. Success lies in the middle, with some important directional adjustments.

The problems currently under attack by the CRE community are outlined in the following blog section:

http://frumpyhausfrau.com/category/cre-loaded/

Be sure to read the comments, there are additional bits of input by other CRE users which outline aspects which I am unfamiliar with (cannot verify), and additional comments by myself in response to questions.

Having already covered what I perceive to be the major problems in my blog entries, I will proceed directly to what I see as potential solutions for some of the current problems:

Primary Revenue Generation Model

It is understood that the previous business model was never quite sustainable. This is going to sound a great deal like it to begin with, except that there are subtle changes. Those changes, combined with the other recommendations in this report, are the difference between “almost right” (that is, unsustainable), and “right” (or profitable and able to build momentum).

I feel that CRE should move from the yearly subscription model, to the following model:

1. **Keep a free version available.** Take out a few of the bells and whistles (built in modules), keep basic functionality on a par with 6.2 in most respects, and make all future enhancements to the free version centered on efficiency within the code base, without further feature enhancements. It is my opinion that the affiliate program feature should not be part of the base CRE offering. It should be reserved for Pro – from both a small business needs standpoint (if they need a free cart they really aren't ready for an affiliate program anyway), and a point of ROI for the user, and value for money, this division makes sense, and it also provides a clear distinction between the three that someone can grasp easily, based on most wanted features. Free is “my store”. Pro is “my store plus affiliates”. B2B is “my store, plus wholesalers”. Require that credits stay in place on site pages for free version, and provide a link in the Admin “upgrade to Pro or B2B” - this upsell is logical, and acceptable, and even helpful to the store owners.

   ALTERNATELY: Provide all new versions at a fee, and the prior version for free, unsupported. This is riskier. It leaves users vulnerable to potential security issues, and does not foster goodwill as much as the first recommendation. It will bring less support to the current project – a lot of people who do their own coding to save money will simply work on the older version instead of the newer one. It is an option, but not the recommended best choice – still better than where you are now though!

2. **Offer the bells and whistles on the Pro and B2B models, with add-on modules as separate purchases**. This is no different, but the value should be defined in short statements for each, that have more meaning to the customers. Currently Pro doesn't offer much of a reason to upgrade for the average small business.
3. **Keep prices fairly low on paid enhancements**. $10 to $20 for a module. People will buy that and hardly notice they paid it – you'll capitalize on impulse buys. I know, my clients do this with other carts that have low priced modules.
4. **Keep prices of the Pro version low enough that people will be able to afford it and feel they might as well purchase it**. $100 seems to be the “breakpoint” for small businesses – they'll justify $100 for software. More, and they look for other options. Allow site owner to remove front-end credits. You'll get purchasers for this reason alone when the price is lower. Pricing here is ESPECIALLY critical with the economic climate we are now in.
5. **B2B should sell for $300,** because it offers a significant ability to increase revenues for the businesses through wholesale orders. More than that, and you'll lose them in droves. Again, with the current economic downturn, which will continue for some time, this is very important. Limit support – but keep that initial price low!
6. **Minor releases (double decimal releases for bug fixes and security patches) free**, major releases with significant feature or performance enhancements should be charged for on Pro and B2B, with the most significant improvements stated frequently. This should be a PUBLISHED POLICY. This way everyone knows what to expect, and it makes sense to them.
7. **Provide better documentation, free, with the Pro and B2B versions**. Documentation recommendations are covered in detail later. Support can be drastically limited if good documentation is available – cheaper for you, better for the customer.
8. **Provide a good manual for $10 with the free version**. Again, small business owners will fork over $10 and not even think about it, but when you push for more, they hesitate more. If the documentation is good, you'll sell a manual to nearly every free cart user, and you'll sell another to every designer who works with the cart. A $10 book will be an easy upsell, and will pay for itself over and over. But it has to be good, and it has to be easy to understand.
9. **Make the source code freely available.** It has been posted on the forum than anyone can request a copy of the code for development purposes, but the fact is, that when one tries, a message appears that they will be notified when it is available! **Excuse me?** Source code is OBVIOUSLY available since the

software has been out for many weeks now. This feels so much like a dodge that there is no way any person who fills out the form will think that it is anything but a shady tactic to avoid making it available.

The point of this is to offer many advantages to the customer, while also providing a model that truly is sustainable. It is a little different than the previous model that wasn't sustainable, for several reasons:

1. **There is a stated policy for requiring additional fees** – People know they will eventually have to pay to upgrade.
2. **There is stated value for payment, both for initial payment, and for the upgrade**. They know you'll charge them when there are goodies they want, but not when it is a bug-fix or security patch.
3. **It complies with standard models within the industry** – it is a model people understand, and will support as complying both with software industry norms, and with the spirit of Open Source.
4. **The lower prices actually enhance sales.** I've watched business owner after business owner choke at even $150, but say, "I can afford $100 for something better, if I can control when I pay it." If you can sell 10 times as many, the lower price makes way more sense. And I believe the difference may be just that.
5. **It leaves all the choice with the customer.** Customers like being given choices, it makes them feel respected. They come back, and are loyal, when you give them the choice.
6. **It puts a "price" that the customer can understand (advertising) on the free version, but allows them to remove it on the paid versions.** Another industry standard. You'll make up more than what you lose in advertising, in momentum from recommendations.
7. **By adopting familiar standards in a fresh way, you can both overcome the problems that left CRE languishing for years,** and spend less time explaining yourselves – because everything is already logical, and happening in the way most users expect it to happen.
8. **By keeping certain things free, and by making source code openly available, you invite the community to come back into a partnership with you**, which enhances the progress and marketing power of the project. This is fundamental to the success of any Open Source project.

Coding and Functions

I am not a coder. Nor am I ignorant of coding issues. I fall somewhere in the gray area between. I can't code my way out of a paper bag. But I understand the theory behind coding, I understand the current coding standards that never occurred to the original coders of OSCommerce, and I understand the nightmare of taking code written by hundreds of different coders without a single coding protocol to guide them, and then trying to combine that code into a single piece of functioning software that doesn't throw someone for a loop every time they try to accomplish a necessary task.

I believe that the first priority from the team should be to make the courageous decision that should have been made in the beginning:

A complete break with the OSCommerce roots.

Before you gasp in shock at that, yes, **I am fully aware of what that means**. I am aware that in every bit of backward OSC compatibility that you abandon, that you lose the ability to move a site forward – you risk losing a long time customer. And I have to say:

Deal with it. Do it. Have the courage to be something more than you can be if you keep that albatross around your neck. A clean break will allow recovery within 1 year. Dragging it along and trying to preserve compatibility will leave you struggling in the dust while newer competition sweeps the field and the negative impact may never leave.

Let backward compatibility go. It won't happen to everything at once anyway. If the users understand that some things simply MUST be done to stabilize and improve the system, they'll accept it. **They accepted it with Joomla.** Many extensions for Joomla simply are not compatible from 1.0 to 1.5. The community KNEW they would not be, as the Mambo code was replaced by newer code. The community also knew that it was worth it, and looked forward to the new version with great anticipation. They were allowed to participate in beta testing, to make reports, and were involved in it. They did try to preserve some backward compatibility where practical, but where it was not, they let it go!

Now, 10 months later, most have moved to the new platform, 90% of the extensions previously available are updated for the new standard, and Joomla is continuing to carve out growing popularity. They lost very few users. Those who just couldn't do it are still stuck in J1.0, but they are gradually making the leap forward, even when it takes custom coding to do it.

Have the courage to let go of the albatross. It isn't giving you wings. It is tying you down.

I recommend:

- Analyze where compatibility issues are likely to be the greatest. Begin announcing that some module compatibility may be lost on the next version.
- Analyze the most popular modules. Make sure compatible versions are created, but ONLY the most popular ones. The rest can catch up in the following year.
- Involve the community in the process – allow extensive beta testing among the community – that way those who wish to begin developing compatible modules early can do so. **No surprises – be open and loud about it!**
- Provide 4 distinctively different, and easy to edit templates with the new version. If the templates are flexible, people will recode their designs and move forward. They'll accept this without much complaint if they know you tried to make it easy, and if the new templates ARE easier to design in. Again, be loud and warn a lot that it is coming, so nobody redesigns their site only to learn two weeks later that the newly released version won't support their slaved over template.

The things I feel are of greatest concern from a usage standpoint:

1. **Separation of editable code from core code.** It is simply unacceptable to have to risk losing changes during an upgrade. Ideally, NO modified files other than config, image and template files, during install and setup. If someone hacks the core because they choose to, that is one thing. If they are forced to do so just to set up necessary elements, or to install recommended modules, that is unacceptable.
2. **Improve database queries** so that response times are faster.
3. **Simplify the backend operations**, so that configuration and setup is as convenient as possible. **I think that THIS is probably the most visible thing you could do to impress people that change was actually happening.**
4. **Consolidate design processes. Quit trying to make bits and pieces easier.** Either have it all in the backend, in ONE PLACE instead of three, or keep it all in the templates and CSS. Don't make me go to 8 different files and three places in the backend just to set my logos, and customize my font colors!

I recommend focusing on those issues to the exclusion of all else, and yes, I realize how many people with their own agenda are demanding bells and whistles and features and this and that! This is the FOUNDATION. Get it right FIRST. **Doing so will set CRE apart from the pack immediately.** It will define it as being superior to every other OSC fork, and it will give it the focus (on performance and usability) and the unmistakable advantage in use that it needs to define itself as different and better.

Your target market says they want backward compatibility. But they want performance and forward progress MORE. Some of them don't know this, only because they cling to the familiar. In this instance, trying to give them both is not only impractical, it will drain resources needlessly, and hamper you from actually achieving the goal of enhanced performance and usability. **Maybe you can have both – but the cost of having both is simply unreasonable, and the time involved is unacceptable. Simplify. Let go.**

Target Marketing

Your target market is not first time do it yourselfers. It eventually can be, but right now IT IS NOT!

Your target market is coders, and designers. People who are familiar with HTML code, and who are familiar with FTP and hosting.

Right now, if you confuse your target market, you will offend and push away the people who buy, and

who will KEEP buying, while you try to court someone whom you are not ready to please. CRE is NOT ready for the complete newbie crowd! That is a long term goal, but not one that is achievable when major performance and usage issues that are annoying for the designers and developers still exist.

Get the foundation right. Then build the finer points of ease of use. It isn't the cart for everybody yet. Focus on making it the best choice for coders and designers. THEN the ease of use for newbies will develop from that.

This is the solution where design issues are concerned also – design NOW, so that it is easy for coders and designers to do the work they do. Give them an easy interface to edit the code, don't make us go to each info-box control in the admin just to change a single heading color! That isn't simple. That is time consuming beyond reason, confusing and backward! Get it simple for the designers, and it will naturally lead to simpler ways to implement later for the newbies to work with.

This is wisdom. Target the right market. Develop for the right market – the low hanging fruit. Be the BEST at what you are good at first. You'll spend less money now, you'll be able to develop more smoothly without fragmenting yourself, and you'll come out with a solid user base that will then grow toward the general population as the features enhance beyond that.

Develop for coders and designers. Market to coders and designers. Right now, that is easy success. Later, it will be easy to expand if you do it in this order.

Your target market is US. That is what we want. We want it to work smoothly and fast for US. **Stop trying to woo a market you aren't ready for, and start trying to please the one you already have!**

Support and Assisted Services

First, get a downloads area back for 6.2, ASAP. This is critical to keeping the trust of your community!

You need open forums. Currently they are not open. They are muzzled. You need community moderators. You have good people, with high standards already there. Enlist their help, selectively. They'll help. Then let the community run itself. Just drop in now and again, and be willing to respond if the moderators pass a message on. Remove people for scamming, personal insults, abusive language, blatant advertising, etc – commonly understood forum rules – your moderators understand that, they can understand rules for banning, and they can handle implementing them. Just basically leave it alone, and it will be fine. It was before. It can be again.

Use the forums to establish company communication with the community. But do not strangle the forums. It must be an environment of open, and informative answers, not just evasions and control for the sake of control. Use it to keep the community informed, show appreciation for regular helpful contributors, and answer questions that can only be answered by staff (moderators can refer those questions to staff). Open communication is your greatest friend here!

These services need to be better quality. Support personnel need to actually be helpful. Paid support must be without major flaws. If it is not, it develops a bad reputation.

Most of your users will never use support, unless you put them into a position where they are forced to. That is bad business. Let them get free help on the forum. Let them purchase an inexpensive manual. Then provide expert paid support help, who knows the stuff you CAN'T get from the manual or the forum.

Documentation

You need better documentation. The manual for CRE has always been lamentable. People would pay $35 for it and then go tell people how little it helped. It has a terrible reputation which all the “testimonials” in the world can't overcome. It fell into the trap of many Open Source manuals – needlessly techie in places, plenty of HOWS, but not enough WHYS, and no help for the stuff beyond the basics that the average person would want and need to do.

The current Getting Started info is a start. But it isn't enough, and it isn't up to date for 6.3. It is obvious someone took some template shots to make it look like 6.3, but it is equally obvious that the guide was written from 6.2, and that the features for 6.3 have not been updated in the instructions.

You need, actually, several types of documentation. All are interrelated, so it isn't like you have to write four manuals. You do need to PRODUCE four manuals, maybe five.

I recommend that ALL documentation cover both 6.3, AND 6.2. No, you are not supporting 6.2 anymore, but people are still downloading and using it. It would take VERY little extra to include 6.2 documentation, but you'd sell many more manuals if you include short notes about the differences. Those extra manual sales, along with the goodwill it will build, would be worth the few minutes of extra time to do it. Low investment, high return.

1. **Beginner's Guide.** This can either be separate, or part of a more comprehensive guide. Just the most basic steps to getting a site going, nothing more. No coding. How to set up basic taxes, shipping, and payments, but no complex stuff beyond the most basic. This should be the first section in a comprehensive user manual.
2. **Designer's Guide.** This needs to present information on customization, more comprehensive info on what the limitations of the cart are, how to install mods, where to find them, and point by point design and config information, and how to use the cart once set up. This needs to be the "go to" book for every single designer who creates a site in CRE. It needs to be priced low - $10, and should be combined with the Beginner's Guide, just for simplification, and to keep people from feeling like they have to choose or spend twice.
3. **Developer's Guide.** Another $10 book, separate from the others, with more technical info. This is where you can give your guidelines. Some happy little coder can write it, so other happy little coders can read it and understand it. Just keep the coders away from the Beginner's Guide, and the Designer's Guide, they don't speak very well to people who don't "get" code!
4. **Store Owner's Guide.** Now, here is a chance to make your mark. No other company is doing this, and they SHOULD! This can be excerpted from the other books, and compiled with JUST the part a store owner needs to add or modify products, add or modify pages, and manage orders, newsletters, etc. No setup, no config, no setting up attributes, etc. Just the part that a designer would need to train a site owner on. Sell it for $35, WITH rights for designers to use it for their clients only. They cannot resell it, they cannot distribute it free to just anybody, they have the right to give a copy to their clients as part of a service contract. Do this, and you will be inviting the design world to come share prosperity with you – you'll be facilitating their business, and developing a partnership with them. By keeping the price low enough that they can afford it without having to consider it very hard, you enable them to get it easily and you save them training time. By keeping the price higher than the rest, you profit more from those who will use it for business, and by NOT selling individual copies, you reduce the likelihood that someone will abuse a single copy license. You have to keep the price low enough that new designers will see it as a useful resource and justify the price though, if you want to sell it where it is most likely to be used.
5. **The Beginner's and Designer's guides need to have special versions produced for the Pro and B2B versions**, unless you want to have separate sections and just produce a single manual. I'd be inclined to do them separately, simply because you do charge for the Pro and B2B software, and the manuals should be available only as part of that purchase, not available to just anyone, otherwise it isn't as much of a value enhancer to the software purchase. **The Store Owner's Guide need not have separate versions** – a single version is fine, because really, what good is instruction on how to use functions in the store after it is set up, without the setup instructions? On the open market, that has little value, so it is safe to release that as a single Guidebook.

These recommendations for documentation are made based on experience as a technical writer (and trainer) who writes for people with no prior understanding of technical issues. So I do not just throw this out with no background for the recommendations. I have written instructional materials extensively.

I believe that manuals – Printable manuals, are your best training resource. They get you the most mileage! They can be printed for reference, used on the computer for fast reference without having to dig out a book, used in CRE Loaded classes (which gets that many more users and sells that many more manuals). They build the most goodwill with your user base, and make you look more prepared, more professional, and more like you'll be there when they need you. There is something solid about books, even an eBook that indicates more dedication than a wiki. They do not require that someone always be on staff, being paid, like support people do, and the writing can be outsourced to a skilled writer for much less than it can be done with a “team” in office. A worthwhile investment.

I think it is ok to password protect (with email address of user) the book. But I do not think it is wise to keep it online (as the previous manual was), or to restrict printing. Either one limits ownership in a way that also seriously hampers functionality. Trust your users far enough to let them use the book fully in honest ways, and you will get fewer dishonest people who abuse it.

Remember that when an individual makes a purchase, they may need to allow their web professional to access their purchased support or instructional resources.

Manuals MUST be updated each time a new version is released. This must be diligently done, and the writer must have access to the software while it is in development, so that the manual version is ready at the same time as the software release.

Peripheral Revenue Generation

I feel that this is one area where the CRE team has fragmented themselves. They've got so many revenue generation schemes in place, that they've lost sight of which is the most important one. I see this all the time with the small businesses I consult with, and the issues are the same for sole proprietors and corporations.

Software sales is your primary revenue generation. Software is your primary product. If you lose sight of that, then the software becomes compromised, and the whole tower comes crashing down!

Merchant accounts are NOT your primary revenue! Put that last screen on the install page, but make it CLEAR that this is an OPTIONAL offering for the convenience of the user! Do NOT just stick it in there like it is part of the install process! That is inherently dishonest, and people FEEL that!

The merchant services in the install are not welcome by many people, mostly because they feel they were tricked. Clean it up. Make it a clear choice.

Templates in the site design area of the admin are probably acceptable, but it should be clearly labeled as to where they come from, and who the credit card bill will be from! Currently it is not, it feels shady and pushy.

The whole CRE Lance site has given many designers a bad taste in their mouth. It has developed a bad reputation among many designers. Clean it up. **It is LOW focus.** It should be there, as an offering, but it MUST not be a source of unfairness to the PRIMARY customer, or your primary supporters (mostly designers!).

When a peripheral income generation scheme causes the company to make decisions that are not in the best interest of the customers and designers who purchase the primary product, then the company becomes fragmented, departments begin competing with each other, and the customer feels betrayed and like the company cares for nothing but the money. This has clearly happened with CRE.

CRELance is a peripheral service. Without the CRE software, it ceases to exist. Whatever decisions are made about it MUST NEVER interfere with the customer satisfaction, or the cultivation of the designer as partner, which is critical to keeping CRE in existence.

Identify what is NUMBER ONE. That is the SOFTWARE.

Everything else takes a back seat. Permanently.

If that change is not made. If CRELance is not managed less aggressively in ways that interfere with the designers who support the software, and more honestly, it will continue to harm the company – currently it does just that, and if you listen, you'll hear more complaints than praises of it.

Summary

Wow, that was long!

I don't have a college degree. I've been in the small business world on and off since I married my husband nearly 25 years ago. I've raised seven of my eight children, and I've learned to see cause and effect very clearly, both from my family, and from the businesses I work with. I am currently a well-respected small business web developer, online business consultant, and Web Development Instructor. I know my stuff, and I've learned it all the hard way. **It is my job, now, on a daily basis, to look at a business, and correctly diagnose what is wrong, and what can be done to put it back on track, with the least possible effort and monetary outlay.**

It is my carefully thought out opinion that the current CRE business model, and policies, are unsustainable. Whether the bad policy came from active decisions, or from the lack of decisions or putting the wrong people in charge of making the decisions, the outcome is disastrous. The company cannot continue to offend customers and alienate the Open Source community, and still stay in business. There is no equivocation in that, this is an absolute statement, made with intent.

Changes must happen if the company is to exist for more than the next 1-2 years. It appears to the users that the funding partners and leadership of this business are acting solely from greed, and from the assumption that squeezing the customer at every point, shutting off every avenue of choice for doing other than purchasing, and driving away those intelligent enough to complain is the preferred method of operating a business.

Harsh words? Yes. This is deadly serious! **This is how the company appears to the community that supports it!** The decision makers need to step back and be willing to trust that goodness has its rewards, and that the goodness of the Open Source community will in turn reward them if they play by the rules that this community expects them to live by.

I have succeeded by going against many common business philosophies. I have given when I could have charged. I've limited my prices when I could have raised them. I've broken out of the accepted norms for my industry that did not make sense for the business I wanted to build – because I didn't WANT to be just another web designer. I know what it takes to be successful in a highly competitive arena, and what it takes to be truly extraordinary. And I know that you get the most, by giving first.

I also know the Open Source world. I started out taking from it, thinking it was a neat idea, and a neat set of freebies. I had to grow enough personally, and financially, to be able to give back. But it happened.

There is no place for vultures in the Open Source world. It is a world full of idealists who want to make the world a better place in a small way. If the CRE Loaded company and team do not demonstrate that same goal, they will fail in the pursuit of profit from Open Source. If they are labeled as vultures, they will fade unspectacularly into oblivion. There are far more failed Open Source projects out there than successful ones which will bear out that conclusion.

You cannot do a halfway job of making changes. The changes must be as deep, and as extensive as the ones listed. There is more than one way of doing this, but it is my considered opinion that these recommendations will give you the BEST opportunity to move forward and KEEP moving forward. If you leave out vital parts – for example, failing to reduce prices, or failing to keep the community involved and informed, then it will fall into a state of half survival like it was in before. It is not difficult to do this. But it is a delicate balance of factors.

My experience and expertise tells me that if these steps are taken, CRE will cease to struggle year to year, and will become something remarkable, capable of winning awards and of being a relied upon and recommended tool for business development worldwide.

Written by Laura Wheeler
Firelight Business Enterprises, Inc.
webtech@firelightbusinessenterprises.com
307-379-2502