

The Client-Developer Agenda

J Kirakowski
Human Factors Research Group
University College Cork
Ireland

hfrg@ucc.ie

Version 1.4

The EMMUS project part sponsored by the EC identified the single most critical issue in successful development of multi-media (MM) applications and web sites as the contact and level of co-operation between the client (for whom the application is being created) and the developer (who is doing the creating and possibly also, if it is a web site, doing the hosting).

At a number of EMMUS technical project meetings, developers, clients, and human factors specialists got together to set down a list of items which should be covered in discussions between a client and a developer at the start of a project. When an agreement had been reached on a short list, we attempted to explain the issues in the light of our experience at HFRG as developers, sometimes clients, and sometimes as mediators or third party consultant experts.

This document has now passed through three rounds of critique and on-the-field testing, and I have attempted in the current version to include all the comments and annotations made to previous versions.

To put it simply, this document is trying to answer the question: *what issues should clients and developers be mutually aware of when entering into a partnership to create a multi-media application or web site?*

I am grateful to many reviewers: from the EMMUS project, from a discussion list on topics to do with professional usability issues, and from our clients and partners. You can get more information about the EMMUS project at <http://www.emmus.org> or from our web site, at <http://hfrg.ucc.ie>. Thanks also to Karen Collins who worked on earlier versions of the list.

Jurek Kirakowski

Contents:

How big is it going to be?
How much content is there?
Who owns the content?
Who provides the content?
Will copy-editing be required?
What graphics work is required?
What will be the main language, and will translation or localisation be required?
Will there need to be contact with 3rd party agencies in order to develop the site?
How clear is the product concept?
What will happen if the client is not happy with the product?
What is success for the client?
After the first version, what is going to happen?
How will client and developer communicate ideas?
How will testing take place?
How much will the lawyers cost?
Apart from the application/ website, what else could a client request?
Who is responsible for security in e-commerce?
How will our web site become known?

How big is it going to be?

For a multi-media application the simplest descriptor of size is the number of computer screens that the application is spread over, although if the application has to do work 'behind the scenes' like searching through databases or doing complex calculations, this is not so true.

For a web site, the best descriptor of size is the number of web pages that have to be developed expressed in terms of number of screens, using a screen size such as 800 x 600 as a standard. Each web page can then be assessed for size in terms of the number of screens it takes to display completely. However, if the developer is going to add a lot of hidden functionality like Java applets or script to the page, the number of these will increase the size of the application. It is worth while making the distinction between applets or script code that have come 'ready made' or easily adapted, and those which have had to be written afresh.

Although the determination of size is important as this is the main cost-driver for the entire project, sometimes, it is difficult to make a realistic size estimate until a good deal of preparatory work has been done. This is a difficult situation both for client and developer. A good question a client may ask the developer is, how many similar such projects has the developer carried out in the recent past, and how does the scope of the current project compare to these? Developers who carry out a particular kind of project routinely will be far better equipped both to cost and to carry out your project. For instance, if you are a hotelier who requires a web site for your hotel, it makes a lot of sense to ask a developer how many hotel web sites have they created in the past 12 months, and can they supply you with the URLs of these sites, how your project compares to those, and so what is the bottom line likely to be.

How much content is there?

This is even more difficult to assess numerically! It is best to keep a running total of the number of image components (ie, how many jpg pictures are there), and the number of words of text. Handling links is a time-consuming activity, and a count of the number of links to be included in the web site is also a useful measure. Care should be made to distinguish between links within the web site (so-called internal links) and links to outside the web site (external links). Links to databases need to be counted separately as well.

These are all quite informal measures of the amount of content that will have to be supplied to make the site successful, broken down between the above categories.

Who owns the content?

It must be remembered that content is not 'free'. Somebody had to create it, and so somebody usually has some right to it. Take care to distinguish between content that says explicitly that it may be freely borrowed and used, and content which appears to be in this category. Check with the site owner before borrowing, and check the conditions of borrowing.

If the client owns all the content there is no problem with regard to ownership, although the legal issue of who will own the digitised version of the content should be settled between the client and the developer or service provider. Some developers claim to own absolutely any content that is part of the web pages they create, and give themselves the right to freely modify or sell the content. Some clients may place severe restrictions on the ownership of their content. Care must be taken in situations when an application such as a web site downloads data from a client-owned database and creates web pages on demand.

Developers may wish to insert a clause in their contract indemnifying them against damages if they unwittingly put up content on the client's instructions which belongs to someone else.

Since a website is theoretically available all over the globe, a developer must also be wary of inadvertently causing an offence by putting up content which may be prohibited in some countries or which breaches local regulations for instance on currency control. A simple answer to this problem is to ascertain where are the main target markets, and to require the

client or content provider to indemnify the developer against damages in those markets; anything else is a shared risk.

Who provides the content?

Ask any developer what the single most difficult aspect of creating a web site or MM system is, and they will tell you without hesitation: getting the content. It is also the single factor that lies outside the developer's influence. It is absolutely essential for the client to understand the need for providing content to match the needs of the site or application. Clients must at a very early stage in the proceedings assemble their 'content provider' team, who is responsible for providing content for the site or application to a rigid time-table.

Failure to provide content is the most easily identified source of friction between clients and developers. Content comes in three varieties:

1. one-off content, that will rarely if ever need to be changed;
2. content that will need to be changed, but at a slow pace (maybe annually)
3. fast-changing content (eg, once a month or faster).

It is useful to review the kind of content the web site or application will handle, and to identify the resources within the client organisation to provide the content at the required time intervals. Realism is important. It may be nice to have an 'employee of the week' page, but are there enough resources to deliver the required information in the required format 52 times a year?

Will copy-editing be required?

The first question to ask is: in what state will the content reach the developer? One may expect by now at least printed text and original photographs and art-work; this content can be scanned in and rendered into an electronic version.

This however is only the beginning. In order to fit the design of the application or web site, the content will need to be transformed by editing. This transformation process can become costly and it should be subject to revision, especially if the content transformers are not domain experts. The client and developer need to know how much transformation of the content will be required to get from the raw content provided by the client to that exhibited in the application,

Content may also need to be 'copy edited' in order to ensure that it is all of the same style and that it projects the desired image. Responsibility for copy editing, if this is required, must be clearly assigned. For most MM and web applications, large amounts of 'black text' are usually inappropriate. A good copy editor will know how to break this material up and distribute it around a page in order to break up the monotony, and to make the information accessible by users whose linguistic skills in the language of the web site are not strong.

What graphics work is required?

Some sites do not actually need much graphic design work, others are intensive consumers of this kind of effort. Some developers have teams of graphics designers who are put onto a new project as a matter of course, other developers assume that most of the graphic design has been taken care of by the client. It is important to assess at the outset how much of this kind of work is going to be required, how will do it, and how much will it cost.

What will be the main language, and will translation or localisation be required?

This is a difficult issue. To some extent, writing content in 'plain English' and relying on visual information to make the point can get over translation problems, but not always. Visitors not only need to be able to read the language of the page, they also need to want to be able to read it. Translation is expensive, and putting web pages in a foreign language can be a disaster if the developer does not have staff for whom the foreign language is their first

language. It is always a good idea to get a native speaker check over translated pages carefully for idiom. Examples of bad English abound and are often the stuff of jokes. This would be poor publicity for your web site.

'Plain English' does not mean limiting your vocabulary or sentence structure. It does mean being careful not to use idiomatic expressions, to take care when using metaphor, and being careful in your use of 'it', 'this' and 'that' when referring to material already covered in the text. As an example, instead of saying 'this issue' when referring to a previous discussion about writing clear English say: 'the issue of writing clear English'. Repetition is fine, especially for readers who are not sure of the language. You should also use one variety of spelling consistently: American or British English. Whenever a list is implied, consider breaking the text into a series of bullet points.

These are some very simple considerations regarding the style of language to be used which may compensate for not going to the expense of getting foreign-language translations.

Will there need to be contact with 3rd party agencies in order to develop the site?

The primary contract is between a developer and a client. Other parties may have to be brought into the contract, such as service providers, content owners, translators, photographers, advertising agencies etc. It is just as well at the outset to spell these out. The more parties that are involved in the work, the more difficult it becomes to manage them and to co-ordinate their efforts. Both client and developer have to ask this question of themselves.

Once set of issues that crops up often is: who will market the product? It is not simply sufficient to put up a web site; someone has to market it to attract visitors to it. Who will do this work? With MM products on CD this is not such a problem, as everyone sees that a product has been created, and that marketing is now essential in order to ship copies. An issue with MM products however is: if the product is not a complete commercial product when it leaves the developer (down to pricing and shrink-wrapping) who will make the product into a commercial package? Client or developer, or is a third party needed with expertise in what is sometimes understood as 'productivisation'?

The cost of making reproductions and packaging should also be considered if the developer is not going to undertake this themselves. If the developer knows who will do the repro and packaging, and what formats are required, much last-minute panic can be avoided.

With web sites, a related issue is, who will host the site once it has been developed? If the developer will be the service provider, the terms and conditions need to be established right away. If a domain name has to be set up and kept 'alive', who will do this, and how much will it cost? A ticklish problem at present is, if there is one developer and a separate service provider, who will own the URL?

How clear is the product concept?

The 'product concept' can be a vague and nebulous thing, and that would be dangerous, because different parties involved in the work can interpret something vague to their advantage. It is helpful to carry out what is known as a 'Context of Use' study at the outset. There is a detailed methodology for this kind of study that can be applied, but in overview, Context of Use means answering three basic questions:

- **Who** is going to visit the site/ use the product?
- **What** will they do once they get there/ acquire the product?
- **Where**, in what technical and social context, will they interact with the product?

The last question includes technical context: ie, the hardware and software platforms on which the product is expected to work, and if it is both CD and web site, how will these components interact?

Clients sometimes say they are aiming their product at everybody, or, they don't know who will find the product useful. The most helpful question to ask in these circumstances is: for each of the above questions (who, what, and where), which are the primary targets, in the sense that hitting these targets will provide the greatest amount of success for the site or product. After the site has been created, it will be possible to do research on the site visitors to ascertain whether the initial client concept is true or not. Luckily, for Information Technology products, the greatest amount of effort is expended in creating the product: manufacture of the finished product is a lot easier than it is for most conventional artefacts of comparable price.

When creating a 'product concept' care should be taken by the developer not to force the pace of ideas by presenting machine prototypes before the client has thought through their initial ideas. Machine prototypes tend to resist change and are therefore dangerous in the initial stages. A typical scenario is that a developer may pursue a machine prototype solution for a good length of time, only to find that eventually the client says that this is not turning out the way they expected it at all, but by then the developer is committed to a particular design and changing from this will give rise to unacceptable costs. Paper prototypes, card sorts, concept walls, and story boards are much more suitable methods for investigating early prototype stages.

What will happen if the client is not happy with the product?

It sometimes happens, either because the client has very precise expectations about the product, or because the client has extremely unrealistic expectations. Most developers have their own styles and methods of creating a product; the client should check out previous work done by the developer and understand the differences between the previous work and how the client envisages the work they are commissioning. These differences must be communicated to the developer before the contract is signed as they may have a serious impact on the development schedule.

Some clients may wish to do a series of rapid test-and-redesign cycles after the bulk of the work is done, to bring the product completely in line with standards, styles, or their vision of the product. It is important for the client to understand how easy will it be for the developer to make changes, and how much will these changes cost.

What is success for the client?

Sometimes, web sites are just 'there'. They simply satisfy a need for the client to own a web site. This is not a problem so long as both client and developer agree that this is going to be a 'vanity' site for which no payback is expected directly (vanity sites are usually created in order to retain market presence so return on investment is difficult to calculate).

Most MM products and nowadays most web sites however, are created to service a particular set of needs in their target audiences. How will the degree of success be measured? Ultimately, this question is answered by economic forces: did the product sell, did the web site lead to increased profits? But this is really the last analysis. Most product owners want to know as soon as possible in advance. There are two ways of assessing how well the product or site is doing:

1. By sales of copies or by site visits;
2. By analysing user/ purchaser satisfaction.

For MM products, (1) is a useful marker, although sales staff are familiar with all the variations of slow starters, fast starters etc. For web sites, (1) is not a very useful guide as the reasons why a site may be visited are numerous, as are the reasons why a site may not be visited. Analysis of the sequence of pages viewed by each visitor is more useful although this is not often a reliable thing to measure and it will not return the reason why the user chose to visit precisely those pages. However, since time is still a very variable commodity on the internet at present, any data to do with amount of time spent by users is not a good guide. Of more use are market studies done on (2), that is, user satisfaction. A good measure of user

satisfaction will return information not only about the extent of user satisfaction, but will provide feedback on how to improve the site for the future.

There are some reliable, numerical methods of assessing end user satisfaction, and many companies now use such techniques routinely to watch how their site or product is competing in the market.

After the first version, what is going to happen?

Clients are sometimes nonplussed by this question, yet it needs to be asked. A web site, even a good web site, changes within six months. It becomes stale. Add to this the consideration that rarely is a web site perfect the first time, and that several iterations may be necessary before the site looks the way the client and the visitors best want it. And then times change, so that information, prices, telephone numbers etc. become out of date. There are therefore at least three reasons why a plan should be made for regular updates:

1. to improve the service offered by the site in the light of client and user comments
2. to update the site in terms of new information
3. to change the look and feel of the site to keep pace with fashion

As a rule of thumb, within the first six months, consider changing a web site at least once if not twice. After that, unless there is a need to post regular updates (eg press cuttings), plan to change or add to it once every six months at least.

MM applications distributed on CD are more variable with regard to timing, and fall more within classic marketing paradigms.

How will client and developer communicate ideas?

A web site or a MM application is closely bound up with the client's image and therefore is likely to raise issues that are difficult to get a rational handle on. The client may insist that they want it 'this way' simply because 'this way' is an expression of the image they want to present. Once an open rift has developed between client and developer on this issue, it becomes extremely difficult to heal. The issue is especially sensitive in that part of the development which is least visible: during the lead-up to the first prototype. Usually, after the first prototype is accepted, development proceeds fairly smoothly.

The use of 'paper prototypes' and 'story boards' is highly recommended at the outset as a way of getting both sides to agree on a common vision, as is competitor analysis.

The most efficient way of establishing a co-operation is as follows:

1. Client makes a 'vision statement' which emphasises the message or the image they want to put across; there may also be some stipulations about colours, icons etc. to match other aspects of the client's image but the emphasis here should be on communicating 'what', not 'how';
2. Developer's graphic designers produce several variations on the theme and exhibit them to the client as paper or screen concepts: there is a temptation to produce minor thematic variants all following the basic pattern - this should be avoided;
3. Client selects one, and usually adds suggestions for detailed modifications;
4. A more detailed paper prototype or story board is developed and tested with the client.

By this stage, the project concept should be fairly firmly established as a first prototype requirement. Subsequent testing will yield feedback. It is important for feedback to the developer to be phrased in the first instance as specific problems that have arisen; the client normally is not in a position to recommend solutions to problems, although sometimes a developer may ask the client to sketch possible solution paths in order to elucidate the nature of the problem.

1. very specific points of detail, known as 'usability bugs', usually static in nature: these should be graded in order of severity from 'show stoppers', to 'inconvenients' to 'minor irritants';
2. interaction issues, usually concerned with mapping the sequence of desired user actions (or task flow) to the implementation of the application or web site therefore dynamic in nature: these can be graded as above;
3. more general issues which apply to many parts of the application, or the overall structure of the application: these are known as 'quality of use' issues and usually require explaining.

How will testing take place?

Testing is a large issue but it can be broken down into four questions:

1. what is the goal of the testing?
2. who will do the testing?
3. what methods of testing will be used?
4. how will the results of testing be handled?

The best way to handle testing is to create a testing plan at the start of the project. The essence of a testing plan is a timeline, from project initiation to, say, 12 months after first release. On this timeline are situated points which correspond to significant technical developments that can give rise to testing activities, such as 'first paper concepts', 'paper prototype', 'first working prototype', 'second working prototype', etc. until 'first release', 'second release' and so on. Note that no recommendation is being made about the number or quality of prototypes to be created: this is very situation specific.

At each point on the timeline, a decision should be made as how the technical development at each point is best evaluated: presentation to client group, expert-based assessment, market testing, formal usability testing, etc., to what depth should evaluation take place, and who should conduct the evaluation.

At some stage before the first release, a technical performance test may need to be carried out, especially if it is a web site that depends on interaction with a large database, or if it is a MM application that depends on a lot of new technology. It is critical that when technical performance tests are being carried out that they conform to the best guesses of the Context of Use analysis with regard to the 'where' question. It is of no use whatsoever to demonstrate high performance of a web site in the same city as the server when most of the target audience is one ocean away.

Agreement should be reached at the outset as to how the results of each evaluation will be handled, that is, who at the developer's site will be responsible for making the changes that are suggested by the evaluation.

The question of who should conduct the evaluations depends on what kind of evaluation is being conducted. Clearly, a review by the client needs no outsiders; but a formal usability evaluation or a market evaluation may need outside expertise if the client does not possess it. It is usually considered a poor idea to locate evaluations solely with the developer, and third-party assistance for any but the most informal of evaluations is recommended. In any case, at each stage during the timeline, the evaluations must be conducted in accord with the Context of Use conclusions (who will use it, what will they use it for, and where will they use it).

How much will the lawyers cost?

Lawyers are required for at least three things:

1. to ensure that the contract between client and developer is a fair one;
2. to protect the client and the developer against claims by 3rd parties (eg copyright, disclosure);
3. to check that the services and content provided by the client are in accord with local laws

It is well worth it, especially if you are a small client company, to get a lawyer to check out the contract your developer or service supplier will offer you. There are many variants, and very little standardisation.

Specialised 'copyright lawyers' may be needed to check out the ownership of key disputed content: it is not recommended to go to extremes here - if the site or application depends on being able to exhibit a particular item of content, and the ownership of that content is disputed, a lawyer may be required to sort it out. However, usually, if an effort has been made to trace the owner with no success, and then later the owner steps forward with proof of their ownership, all that happens is the site or application owner takes off the disputed material if an accommodation cannot be reached.

Don't forget to have the usual disclaimers on your site or product about accuracy and liability: see the small print on any software CD for an idea of how much liability you can disclaim! However, web sites which have a large and detailed page about legal liability and disclaimers are usually regarded (at least at present) with some suspicion.

Apart from the application/ website, what else could a client request?

A useful addition to the service a developer provides would be an annotated or even better a documented listing of the source code that created the application or applets used. Even if the same developer will be used later, such a documentation is invaluable.

Most web sites and MM applications are designed to be usable from the first, but occasionally, it may be necessary to request a user handbook from the developer. Clients may wish to re-edit the handbook if they are going to pass it to their customers, to ensure that it meets the style requirements of their organisation. Most e-commerce sites and MM applications should not require a handbook - they'd be dead if they do, but a 'help' page or menu item may sometimes be useful. User testing is the best yardstick of this. Note, it is preferable to make screens self-explanatory than to provide garrulous help messages. Sometimes, especially with a MM application, it may also be useful to supply a help line on technical issues that the client would not normally be experienced in (eg internet connections, software components).

Some complex sites and applications may also require training in how to maintain, add to, or service the software if this is not going to be done by the developer: but here we are beginning to move into complex issues and clients who have to cope with such issues usually have the technical background and expertise to negotiate a good deal with the developer on them.

We are now getting into deep waters indeed, but if your web site changes often, or gathers data or statistics from users, it will need to be backed up on a regular basis. A simple site consisting of pages which change little from one week to the next does not require backup. What is backed up and how are issues the client will have to discuss with their service provider. Again, clients who need this level of service ought to have experienced personnel in their employment who will be able to negotiate effective solutions.

Who is responsible for security in e-commerce?

There are two main security risks in e-commerce:

1. disclosing personal information about your visitors that you need to complete the transaction but which may be harmful if known by malicious people: for instance, presenting information on the current whereabouts of a visitor, or what your visitor has purchased;
2. lack of security in keeping confidential matters which ought to remain confidential.

Security risks of type (1) are so many and varied that it impossible to advise in any meaningful general way. The less your web site posts about individual visitors the better: the classic case was a golf enthusiast's site which published individual handicaps, dates, and times; thus golfers who were away on holiday could have been advertising their lack of

presence at home to all and sundry by entering their latest handicaps on the courses they were visiting.

Security risks of type (2) are beginning to be tackled seriously. One should avoid any solutions which are generated by a lone developer. In the area of security and encryption, tested public methods are always better than one-off applications. Even so, security risks abound. For instance, a web site which sends a confidential security password as an open-text mail message to the recipient, or an application which stores credit card details of all its past clients in a server attached to the internet. If your security is breached (eg, a hacker gets in through your security layers and extracts sensitive information about your visitors) you may not be blamed, but you may find that confidence in your site drops.

The area is so complex at present that if you are going to handle sensitive information over the internet, you are well advised to hire an independent security advisor who can audit your e-commerce operations independently of the developer and service provider.

How will our web site become known?

This is a key question for a web site owner. There are essentially two approaches to this, and it would appear that client and developer each have a role to play in both.

1. ensure that the web pages have got good 'meta' descriptors of their content, and that there is enough useful text in the page headers to allow a useful description of the page if summarised on a search-engine. Register the pages with search engines. Sometimes, this incurs a cost, and it is always an activity that ought to be done periodically, about every six months.
2. make a list of web sites that ought to have links to your site, and web sites which offer the same kind of material as your site. Write to both kinds of sites, and invite them to create mutually-reciprocating links. Identify sites which contain site listings in your domain, and invite them to include your site.

Both kinds of activities require a modification of the site after a certain time, so it is advisable to include a milestone in the time-line when this should happen. Making a web site known is not a job that can be left solely to the developer or the client, and after the site has been launched, is most probably the most important single next step.

Advertising a site, however, is much the same as advertising any other commodity: non-electronic methods at present are if anything even more important than internet-based methods. Thus adverts in trade magazines, posters, leaflets, and surface mailing are all contributory activities. If the web site has developed a certain style and appearance, it is useful to follow this theme through in promotional literature: the developer should be aware that their designs may also be used in printed media.